rel 9/11/94

p - 238

Final Report

on

PRE– AND POSTPROCESSING TECHNIQUES FOR DETERMINING "GOODNESS" OF COMPUTATIONAL MESHES

Contract #NAS8-38478

National Aeronautics and Space Administration
Marshall Space Flight Center

TR-93-06
April 1993

Computational Mechanics Company, Inc.
7701 North Lamar, Suite 200
Austin, Texas 78752
(512) 467-0618

COMCO

In this section, we present numerical examples illustrating the effectiveness of the approach outlined above.

Example: Stokes problem with smooth solution

As a first example we solve Stokes problem when the true solution is known explicitly. While this is hardly a realistic problem, it does allow us to compute the true error exactly and thus be able to assess the performance of the error estimator. The true solution was chosen to be

$$\begin{aligned} u &= \cos x \ \cosh y \\ v &= \frac{1}{2}x^2 - \sin x \ \sinh y \\ p &= \nu y \end{aligned} \tag{157}$$

and the Stokes equations were solved on the unit square with both inflow and traction boundary conditions. The results obtained are shown in Table 3. The two types of estimators shown in the table correspond to two strategies for obtaining local problems from the global statement (154) (see [25] for full details). The results obtained indicate that the error estimators perform reliably.

Example: Stokes' Approximation to Backstep Problem

The second example is the familiar backstep channel problem modeled using the Stokes' approximation ($\nu = 0.01$). The geometry and boundary data are shown in Figure 3.25.

Of course we cannot compare the estimated error with the true error in this case because the true solution is unknown. This time we have used the error estimator to solve the problem adaptively giving the sequence of meshes and local error distributions shown in Figures 3.26 through 3.34. Physical intuition indicates the satisfactory behavior of the error estimator.

Example: Oseen's Approximation to Driven Cavity Problem

In order to study the effects of a convection dominated flow, the Oseen approximation ($\nu = 0.01$) to the driven cavity problem is analyzed. Here, a standard driven cavity problem has been taken and a rotating convective field $\boldsymbol{b} = (y, -x)$ has been superimposed (see Figure 3.35).

Once again we cannot compare the estimated error with the true error in this case because the true solution is unknown. The error estimator has been used to solve the problem adaptively giving the sequence of meshes and local error distributions shown in Figures 3.36 through 3.43. The performance of the estimator once again appears satisfactory in spite of the strong convective effects present in the problem.

Summary and Conclusions

We have outlined a procedure whereby reliable *a posteriori* error estimates can be constructed for *h-p* finite element approximations of broad classes of problems. The basic approach is the same for all types of problems and therefore is suitable as a basis for a general quality assessment package incorporating uniformity of approach with reliability and robustness.

The situation regarding solution enhancement is much less obvious. This is reflected in the literature where work on postprocessing of solutions of many classes of problems is either scarce or non-existent. For some classes of problems (such as linear elastostatics) very effective procedures are available. However, the approaches are extremely problem dependent.

Faced with the problem of developing a general solution enhancement package, it is necessary to choose whether to require that the user input the type of problem being solved and to implement those procedures which are know, or, alternatively to attempt to construct a general approach. The advantages of the latter are obvious, but these must be carefully weighed against the disadvantages. In particular, it is doubtful whether any single approach can perform as well as the tailor made algorithms and, indeed, may even perform extremely poorly. If this latter course is to be adopted then the most promising area seems to be based on the element residual method.

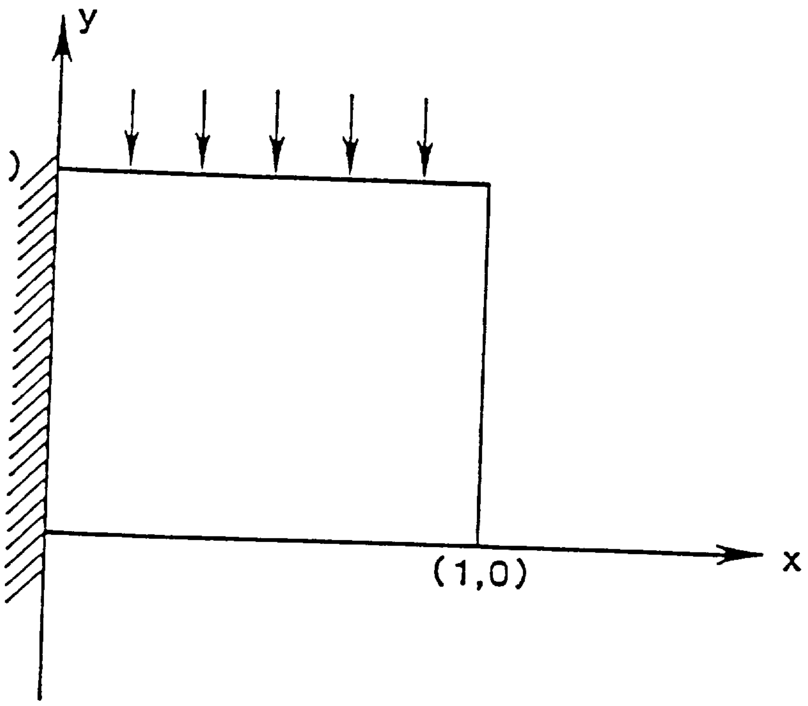


Figure 3.13: Geometry for short cantilever problem.

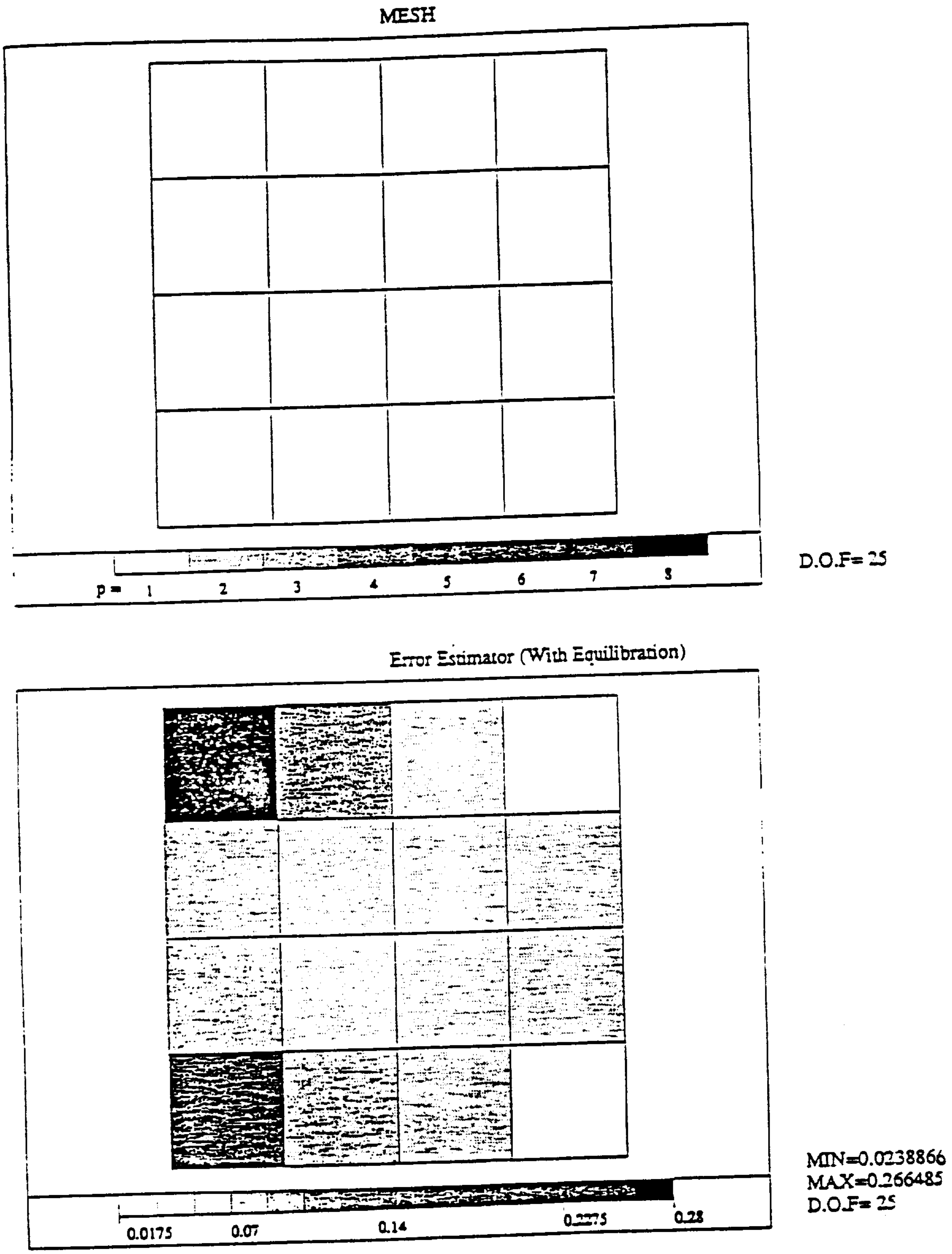


Figure 3.14: First mesh and plot of error estimates of resulting solution for the short cantilever.

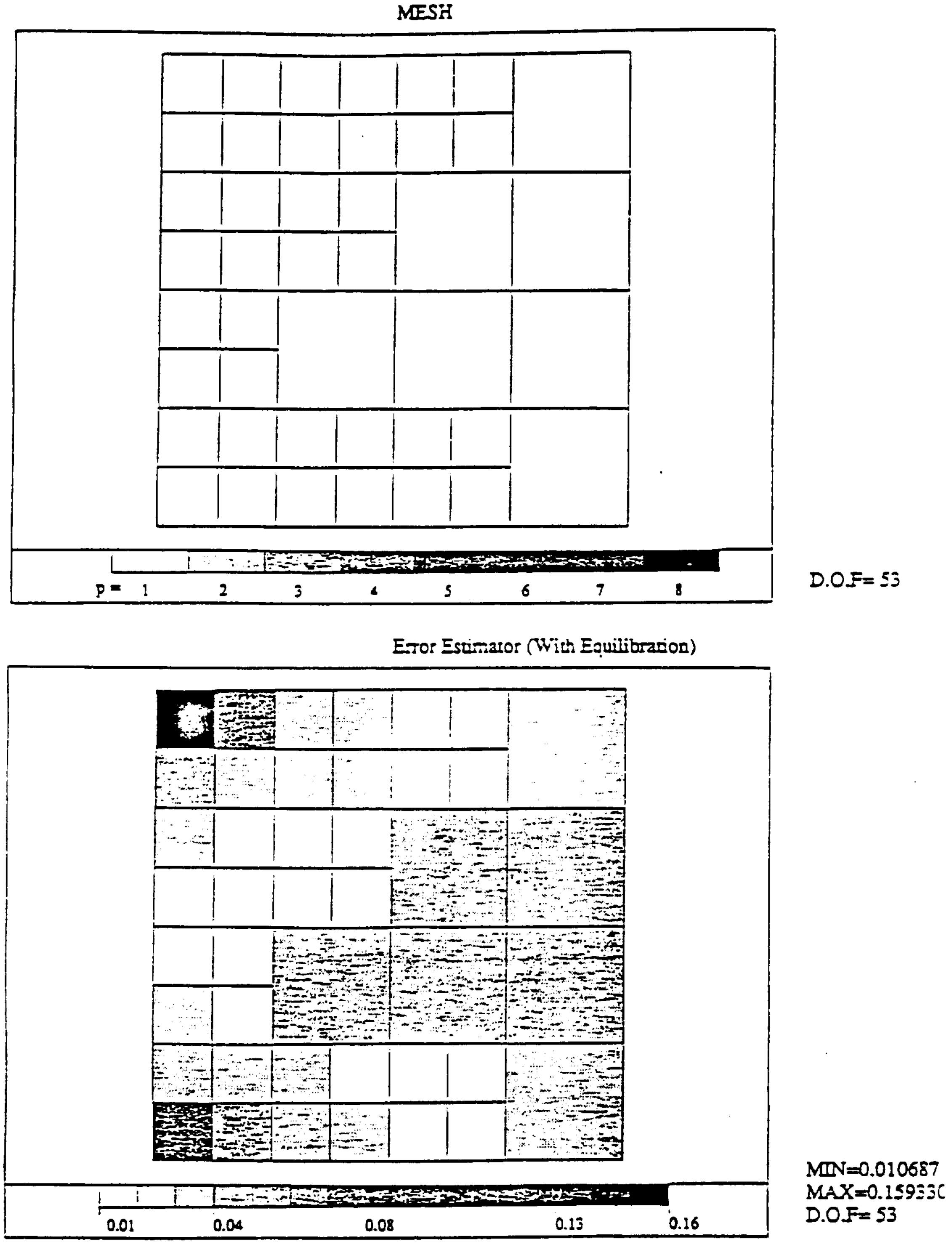


Figure 3.15: Second mesh and plot of error estimates of resulting solution for the short cantilever.

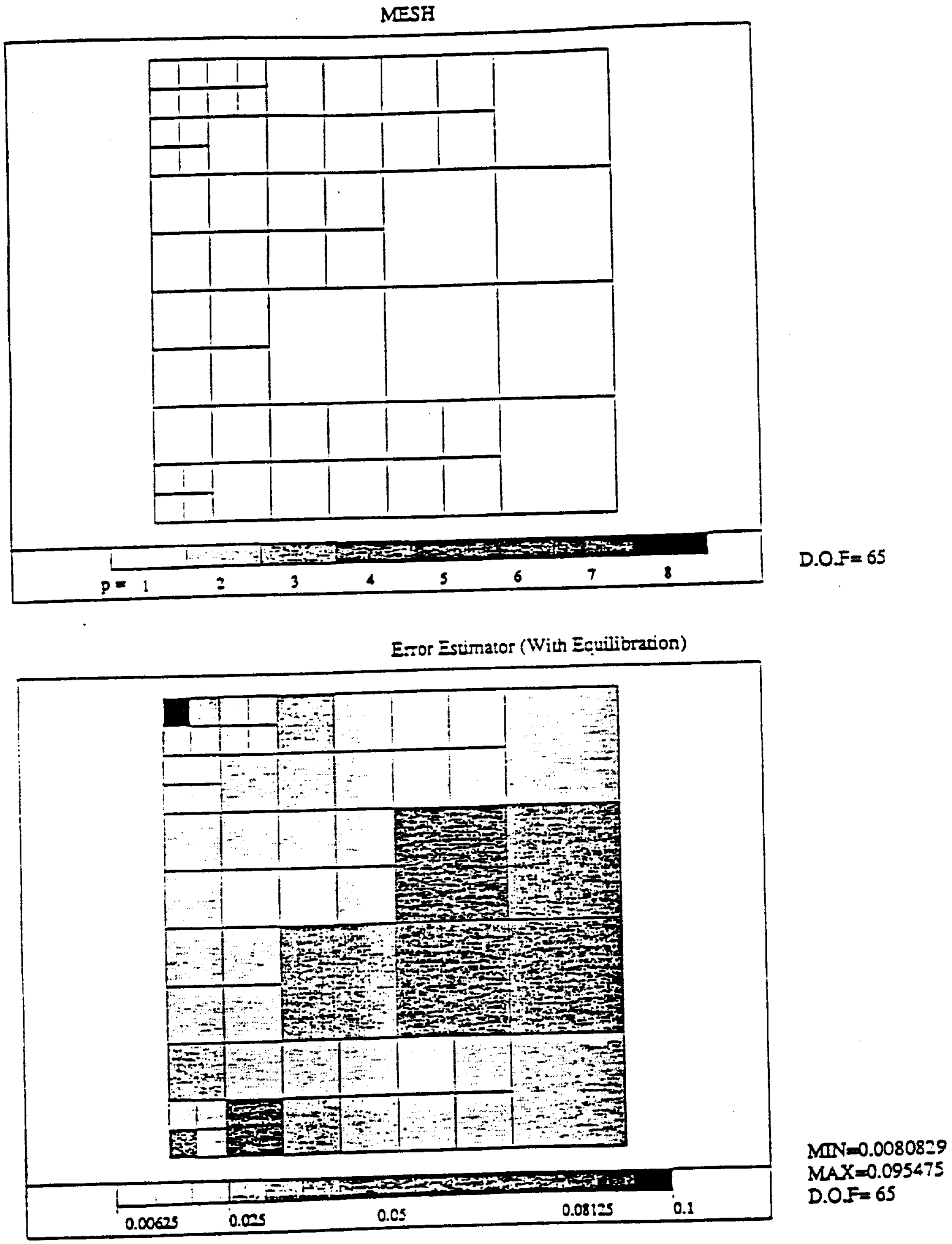


Figure 3.16: Third mesh and plot of error estimates of resulting solution for the short cantilever.

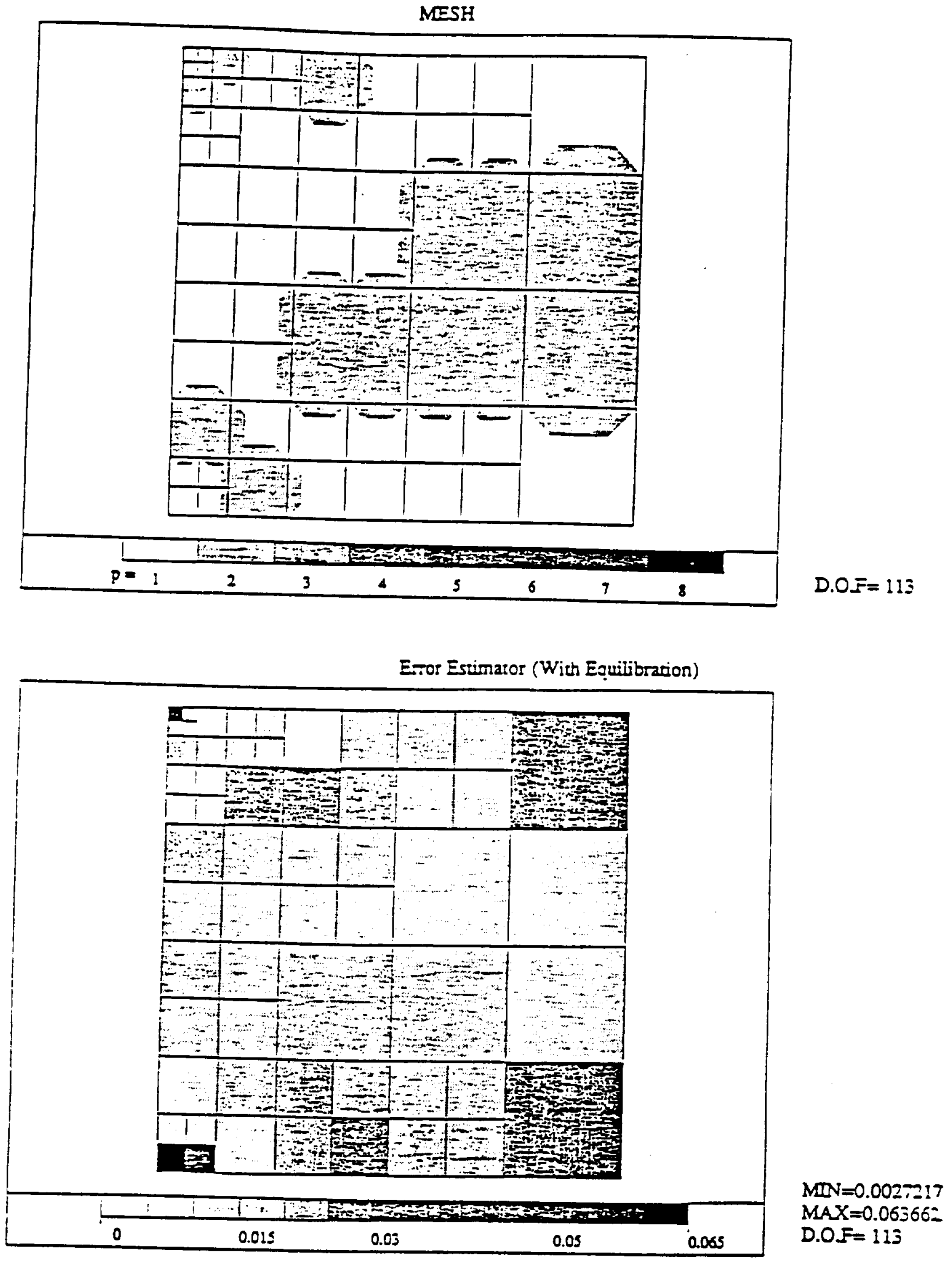
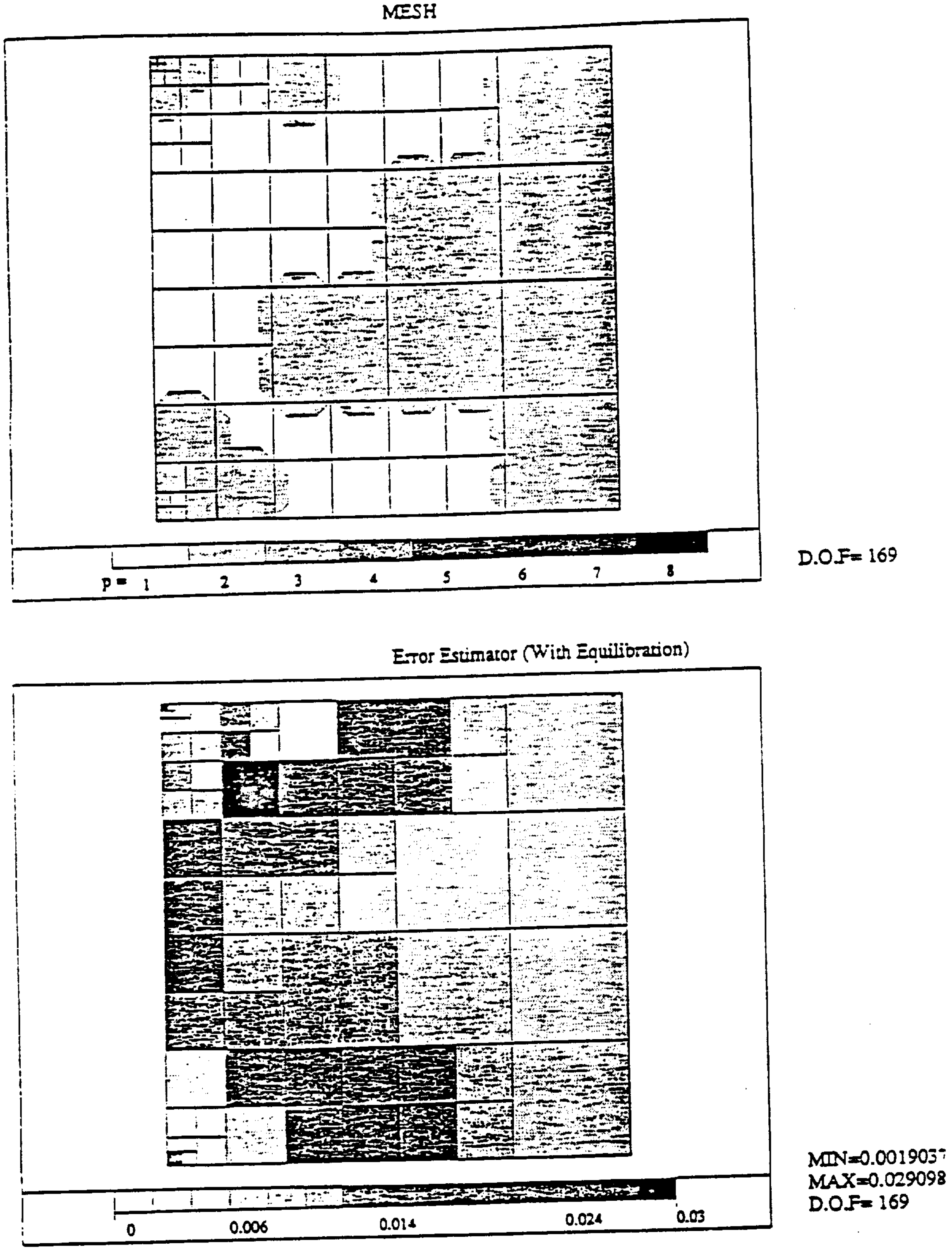


Figure 3.17: Fourth mesh and plot of error estimates of resulting solution for the short cantilever.


MESH

Error Estimator (With Equilibration)

Figure 3.18: Fifth mesh and plot of error estimates of resulting solution for the short cantilever.

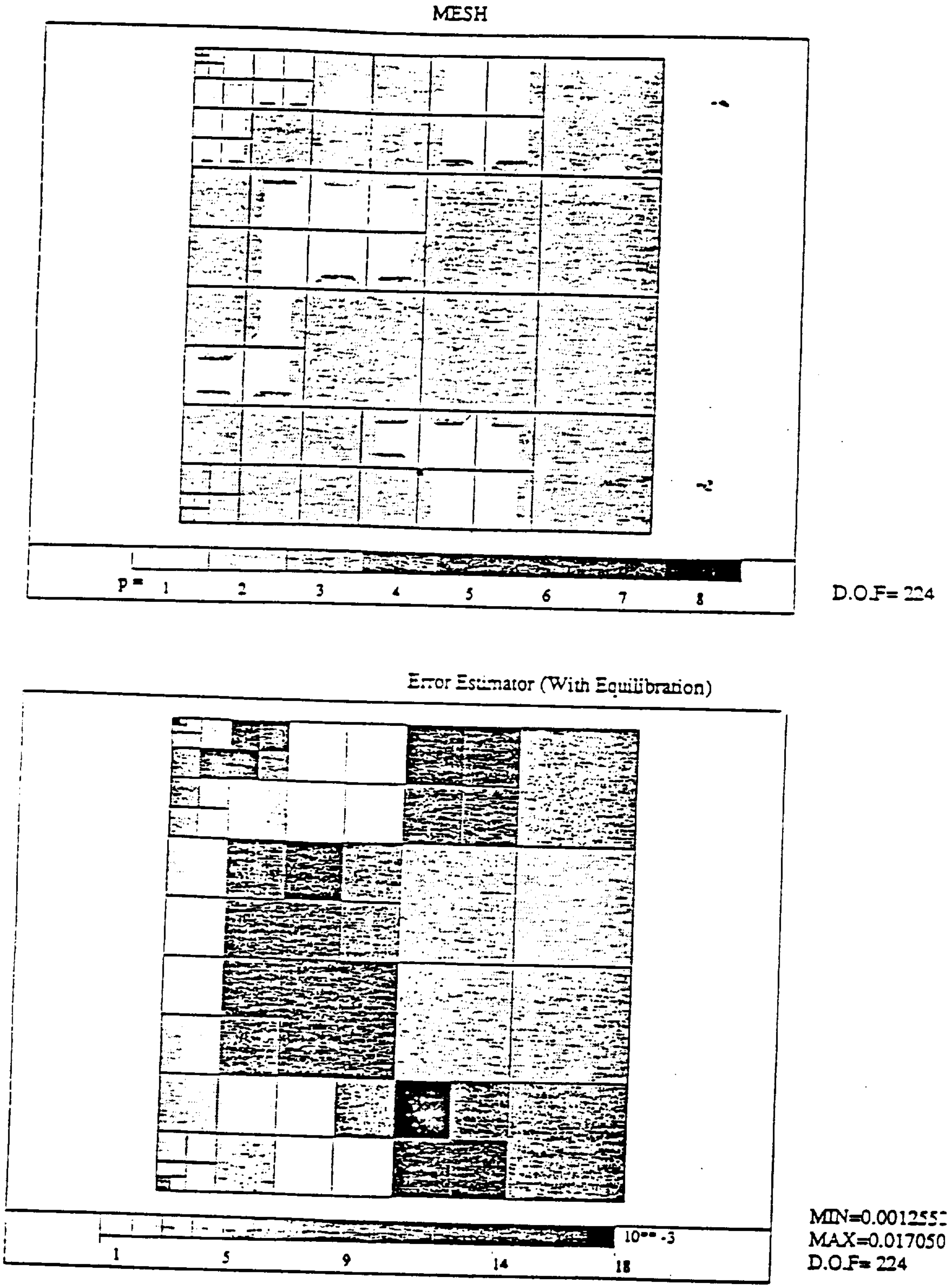


Figure 3.19: Sixth mesh and plot of error estimates of the resulting solution for the short cantilever.

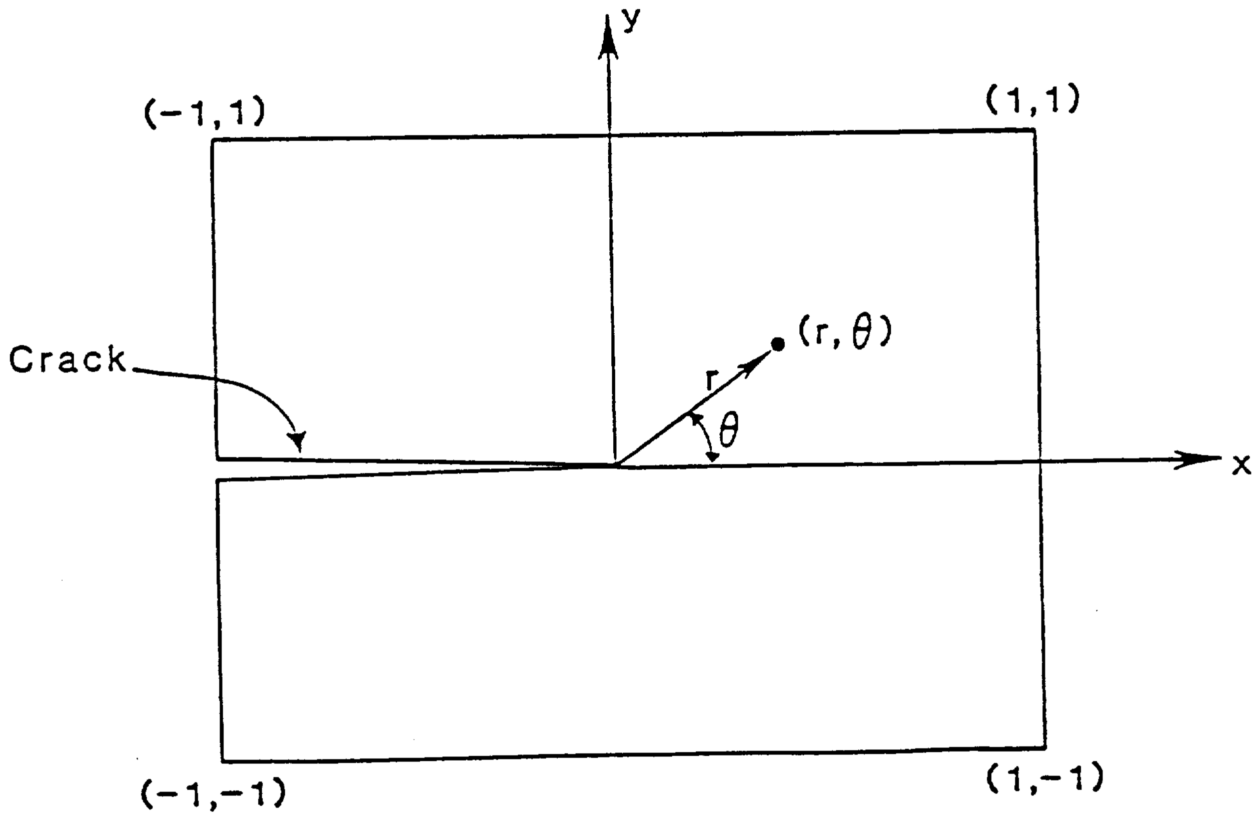


Figure 3.20: Geometry for cracked panel problem.

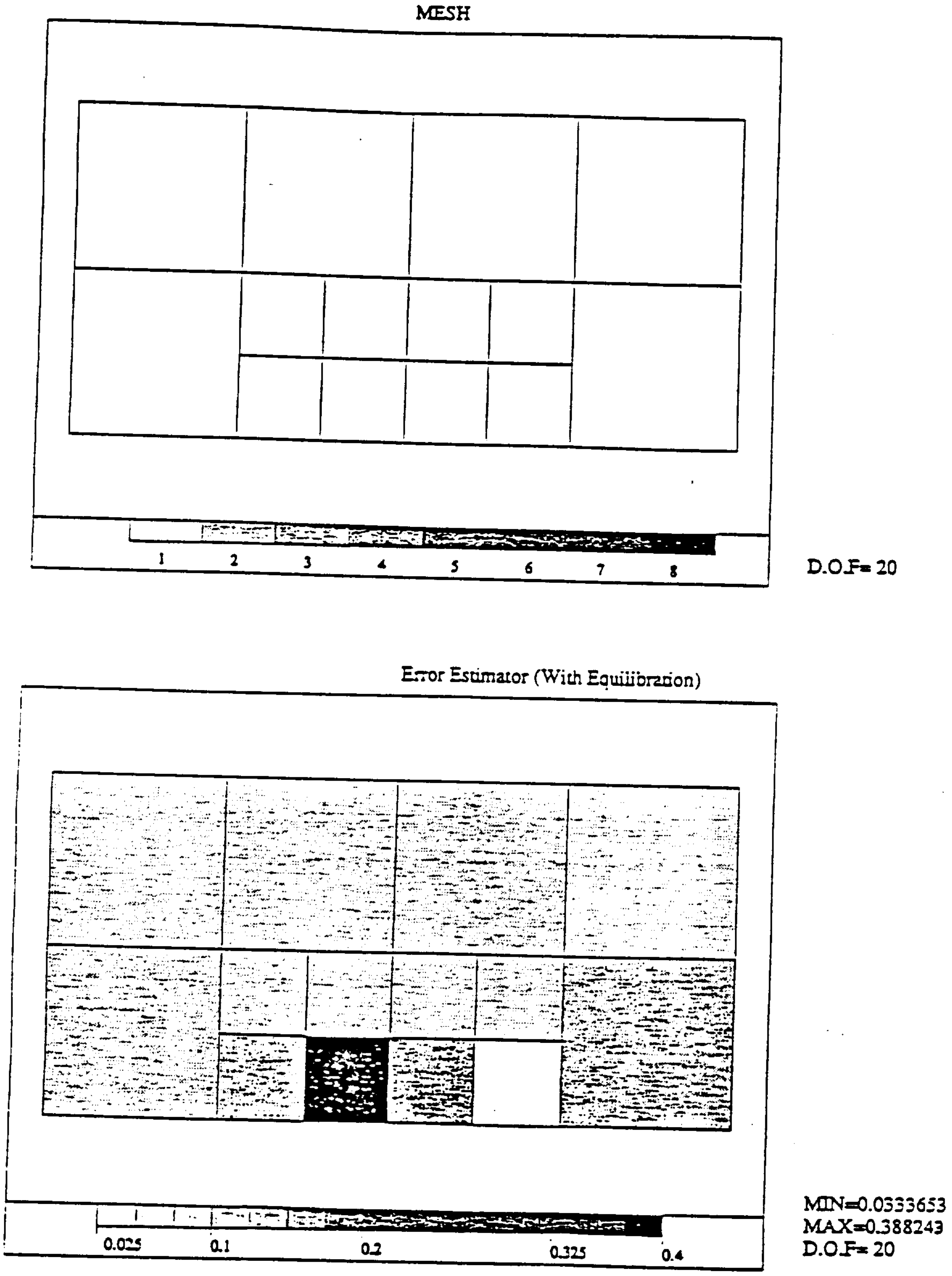


Figure 3.21: First mesh and plot of error estimates of resulting solution for the cracked panel.

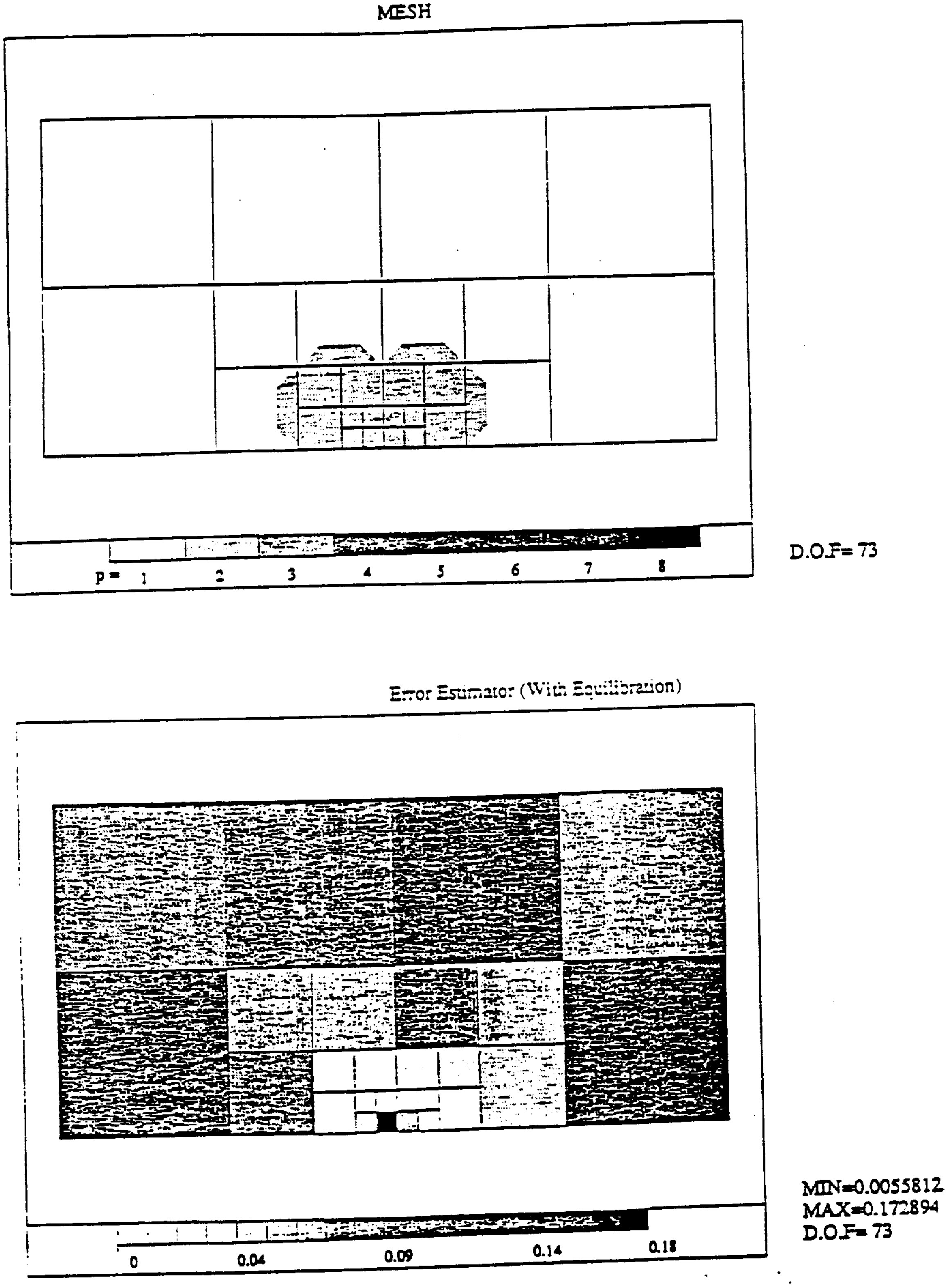


Figure 3.22: Second mesh and plot of error estimates of the resulting solution for the cracked panel.

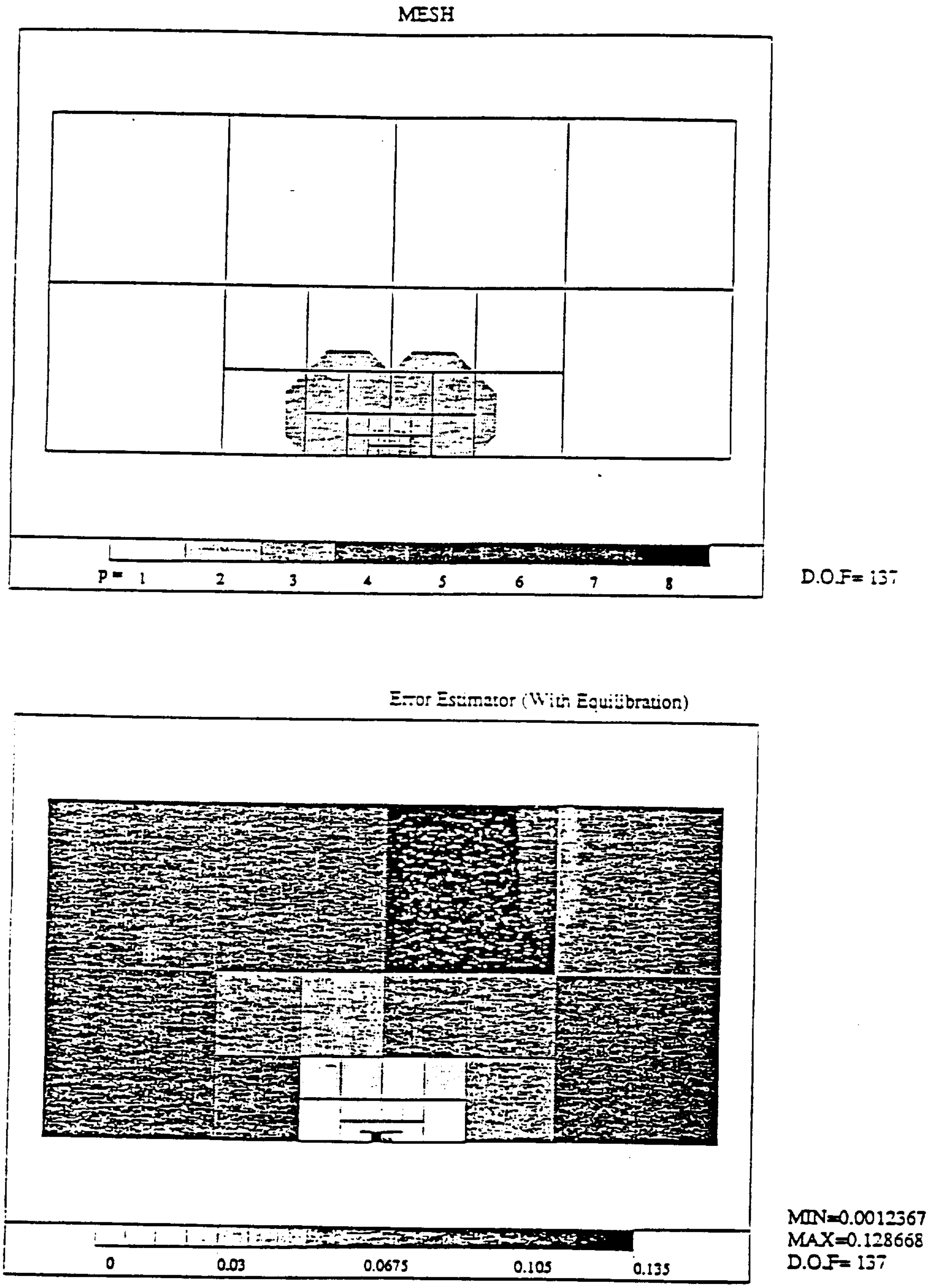


Figure 3.23: Third mesh and plot of error estimates of the resulting solution for the cracked panel.

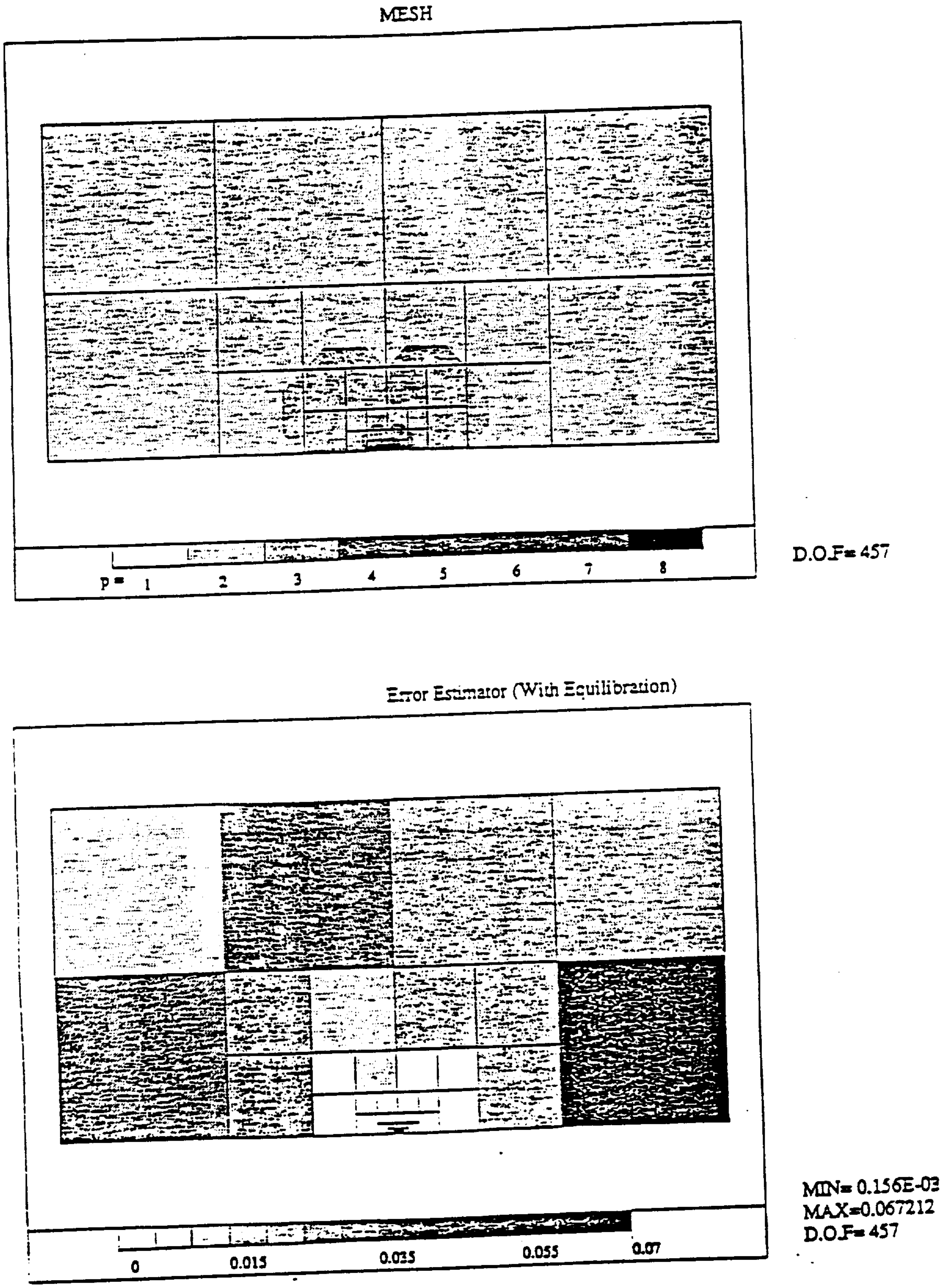


Figure 3.24: Fourth mesh and plot of error estimates of resulting solution for the cracked panel.

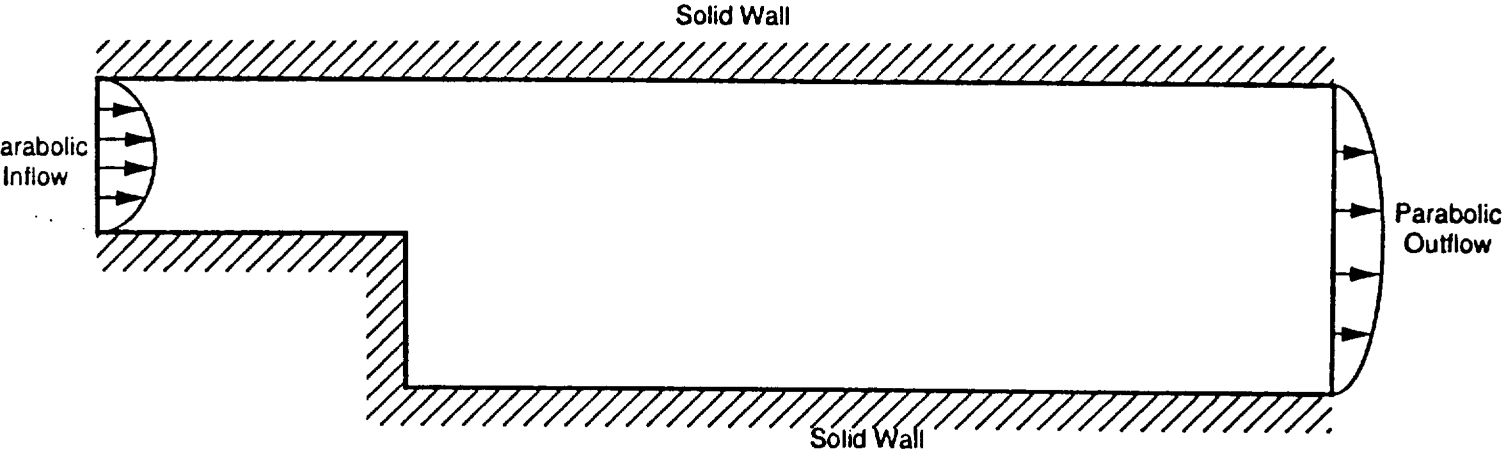


Figure 3.25: Geometry for the backstep problem.

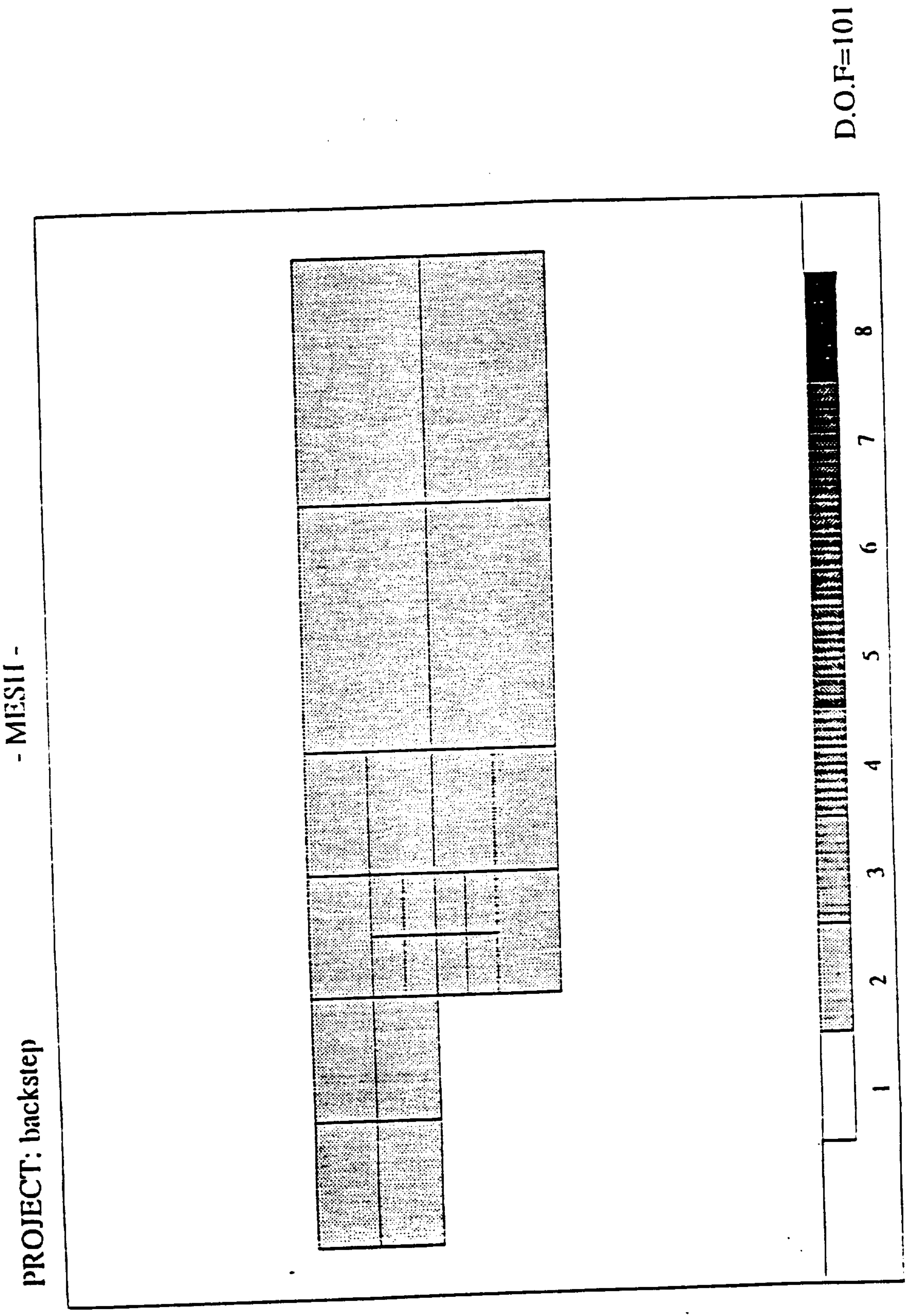


Figure 3.26: First mesh for backstep problem.

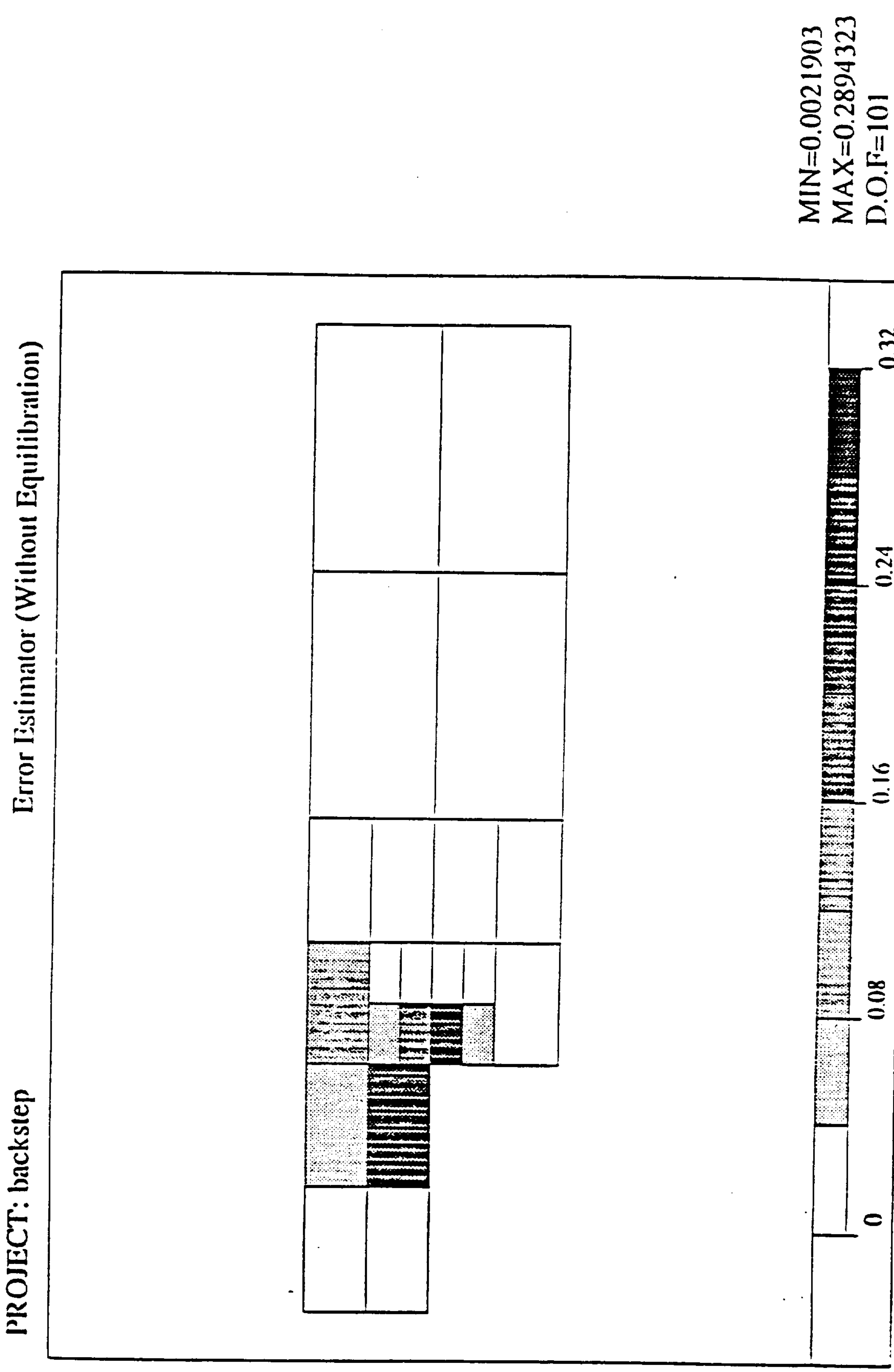


Figure 3.27: Plot of error estimates of solution on first mesh without using flux equilibration.

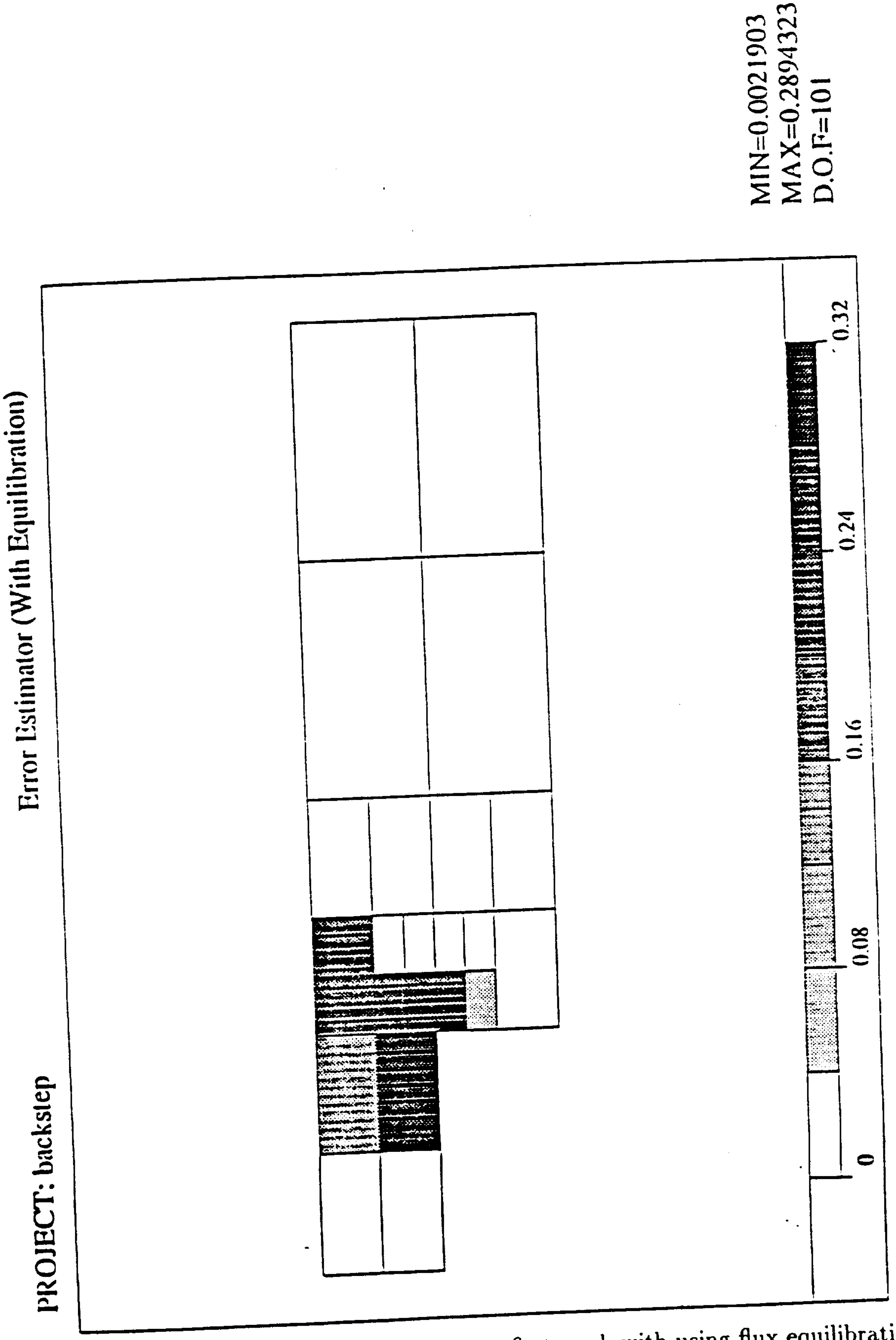


Figure 3.28: Plot of error estimates of solution on first mesh with using flux equilibration.

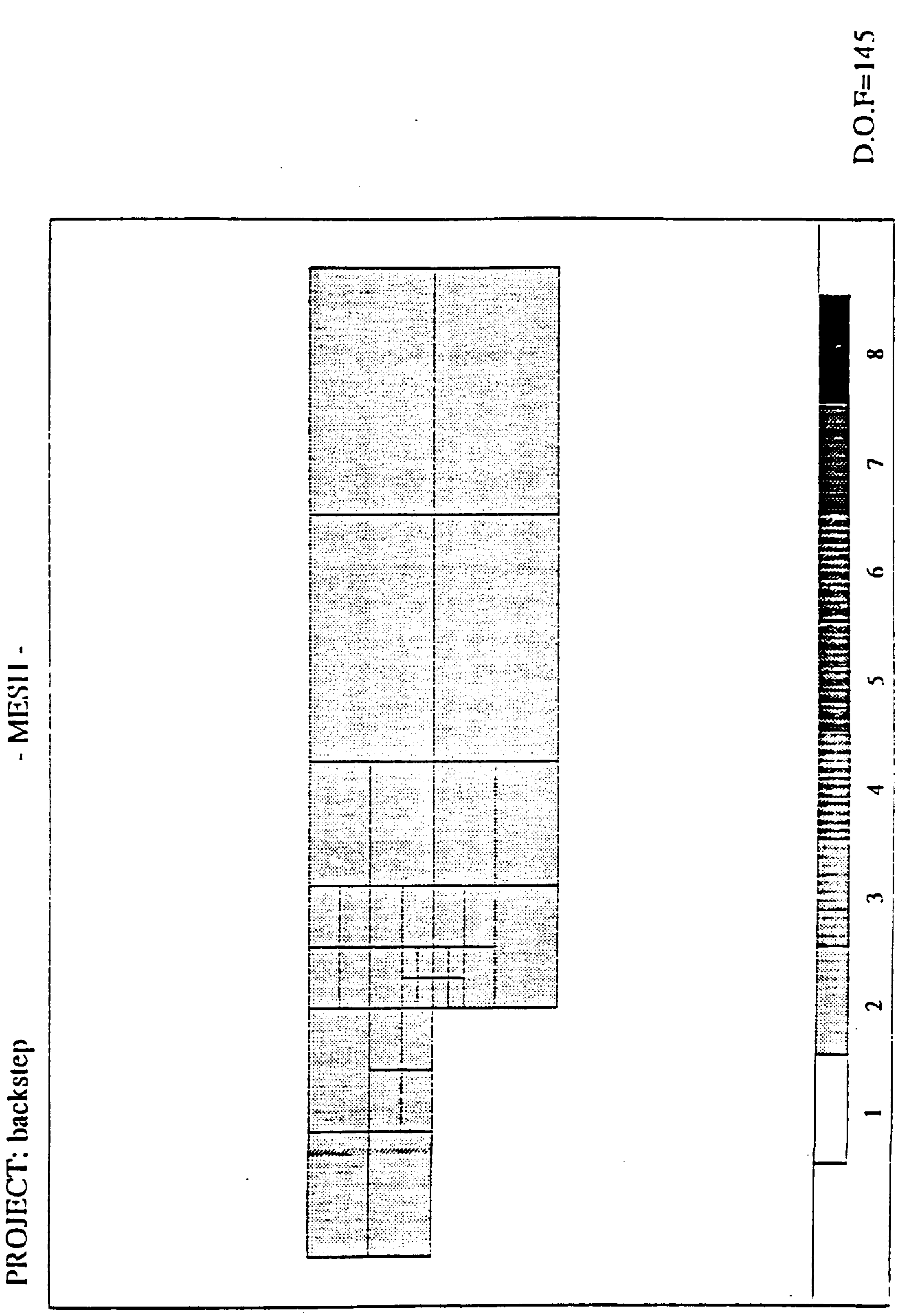


Figure 3.29: Second mesh for backstep problem.

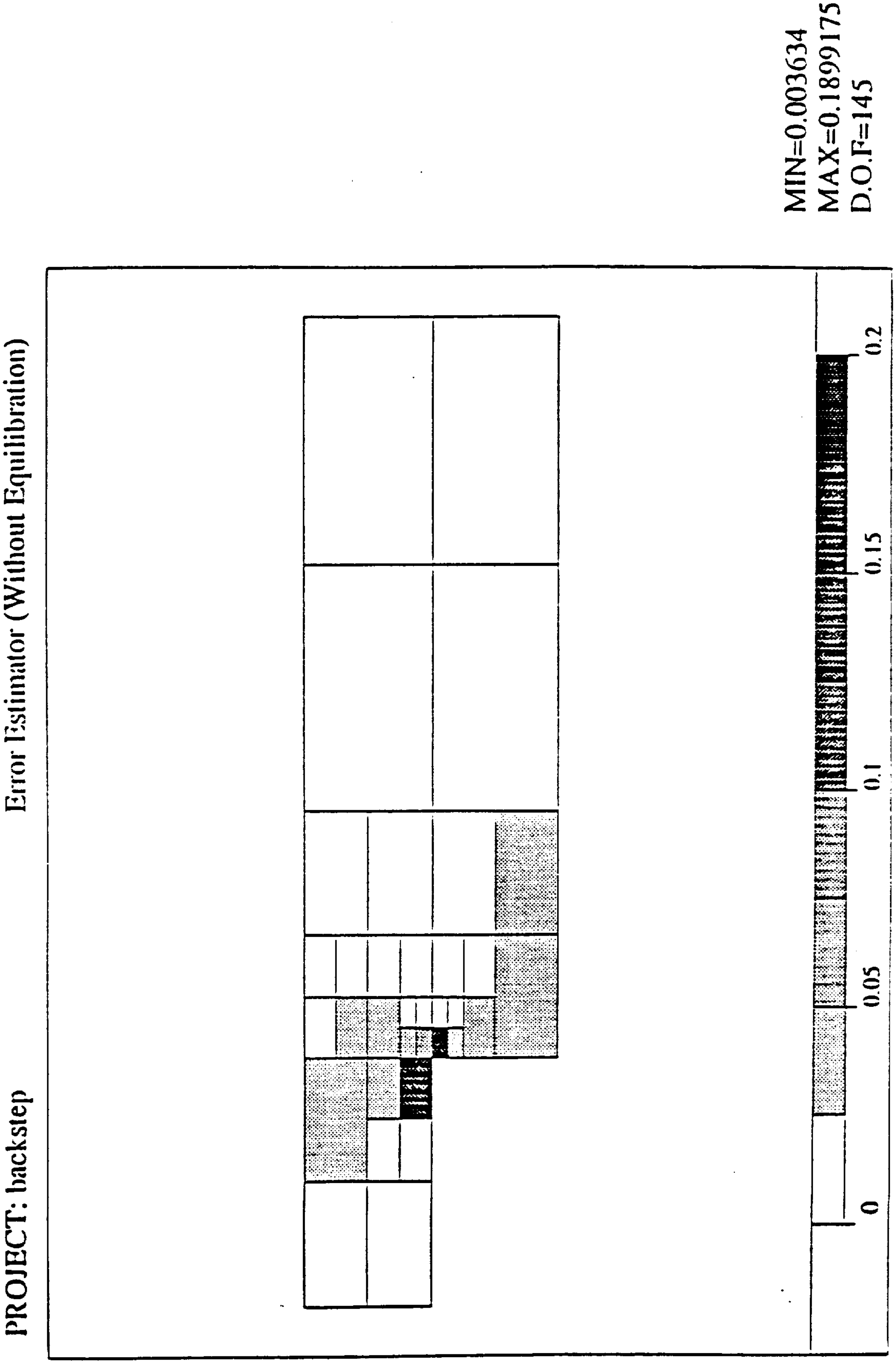


Figure 3.30: Plot of error estimates of solution of second mesh without using flux equilibration.

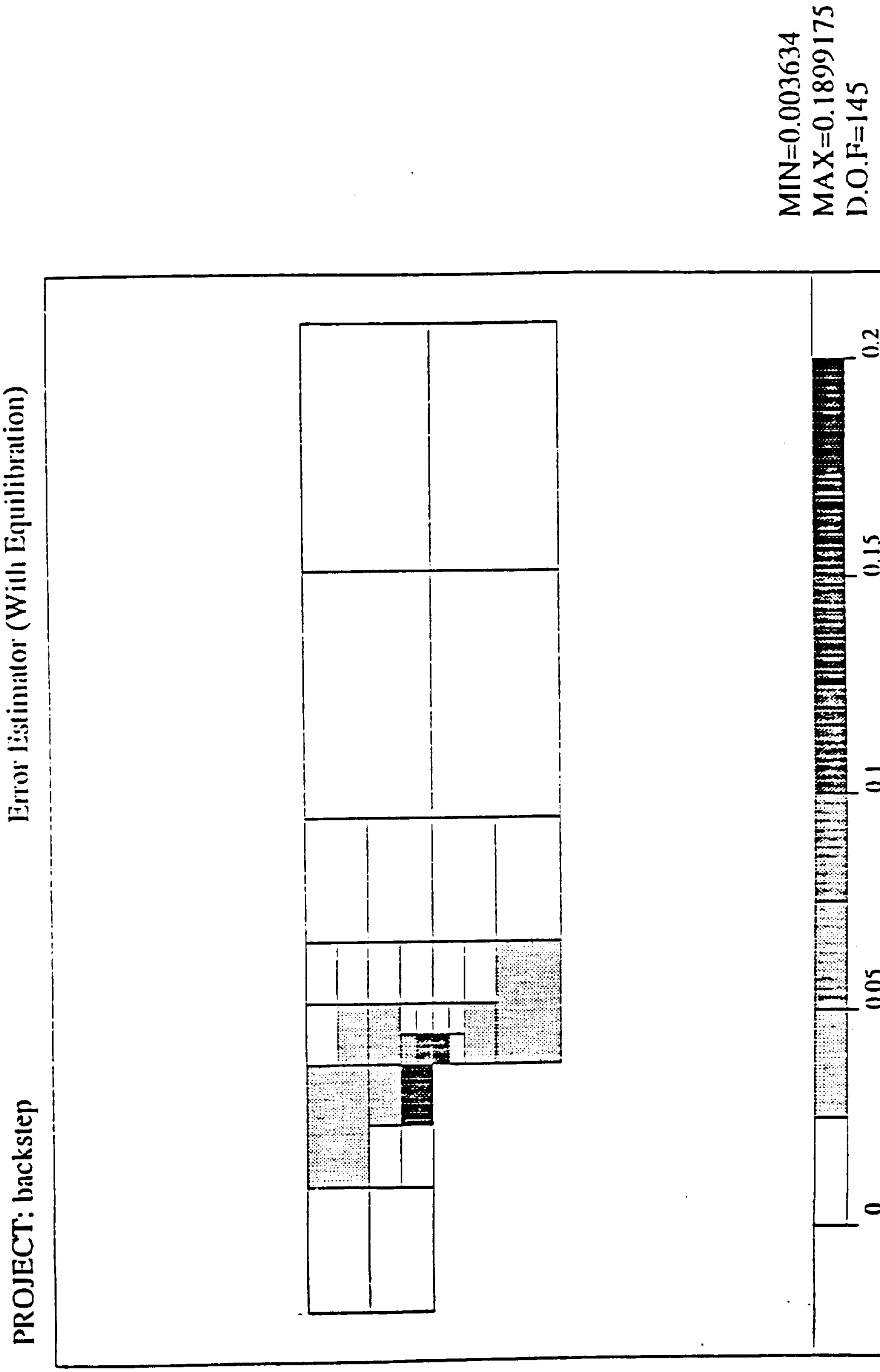


Figure 3.31: Plot of error estimates of solution on second mesh with using flux equilibration.

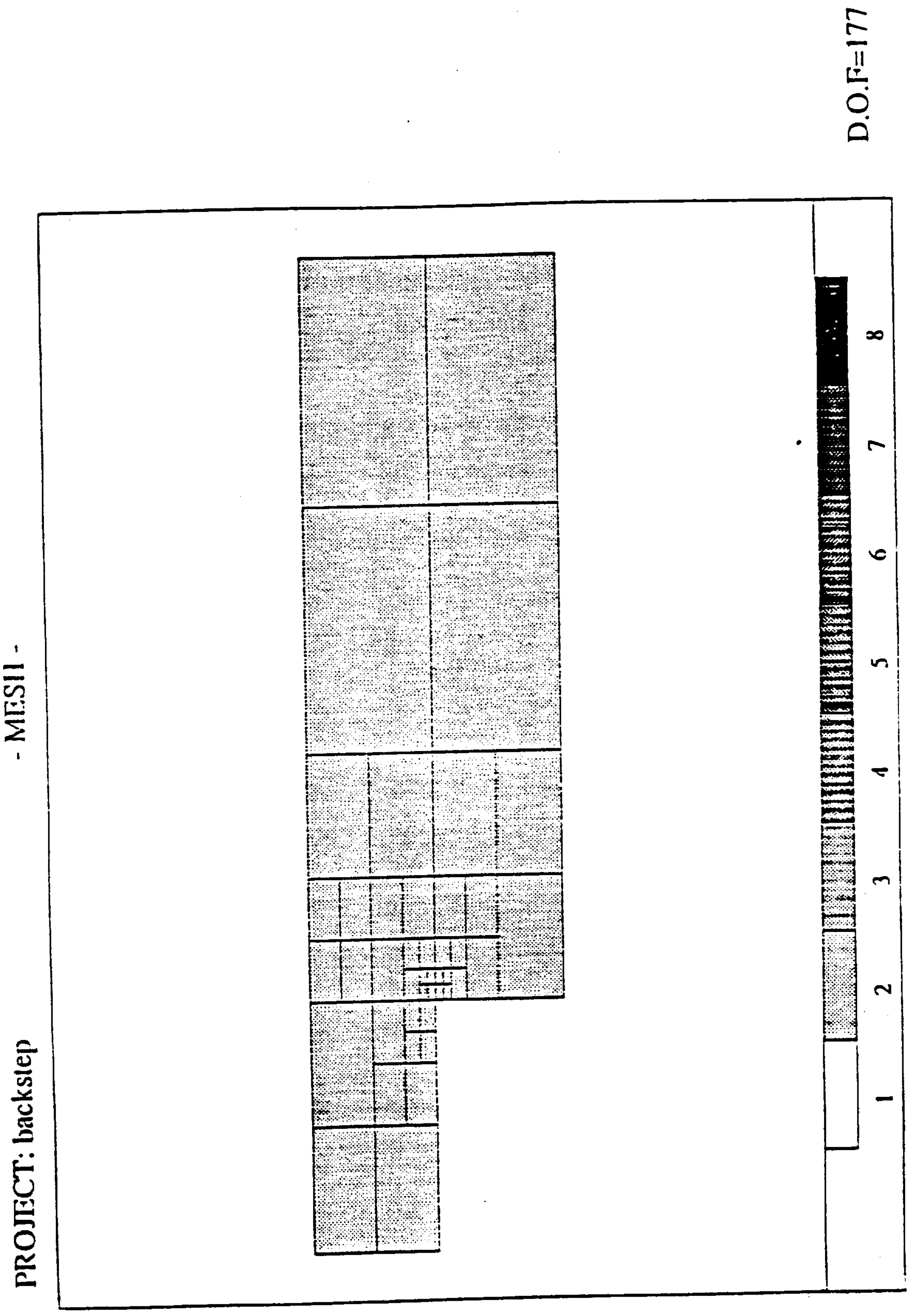


Figure 3.32: Third mesh for backstep problem.

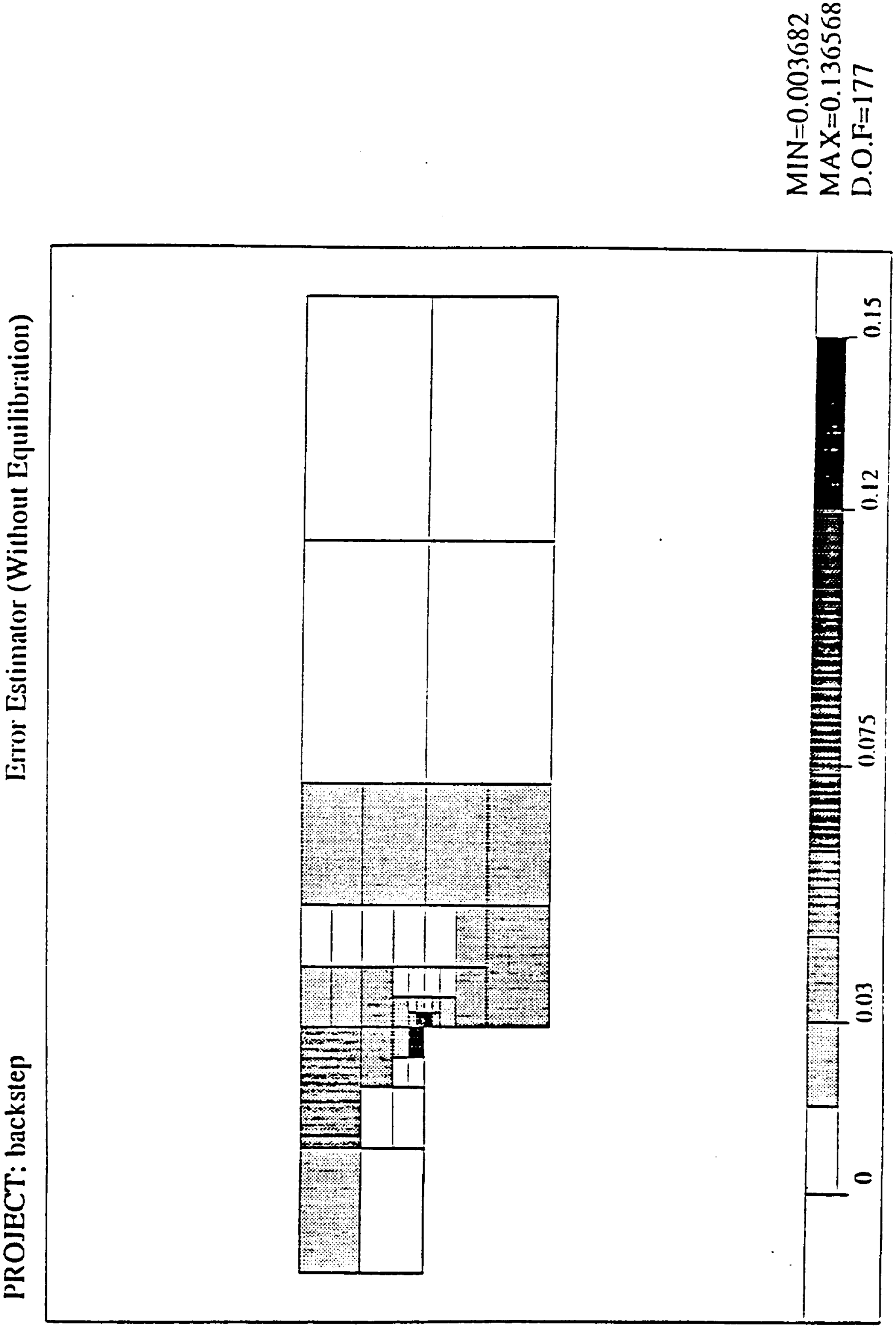


Figure 3.33: Plot of error estimates of solution on third mesh without using flux equilibration.

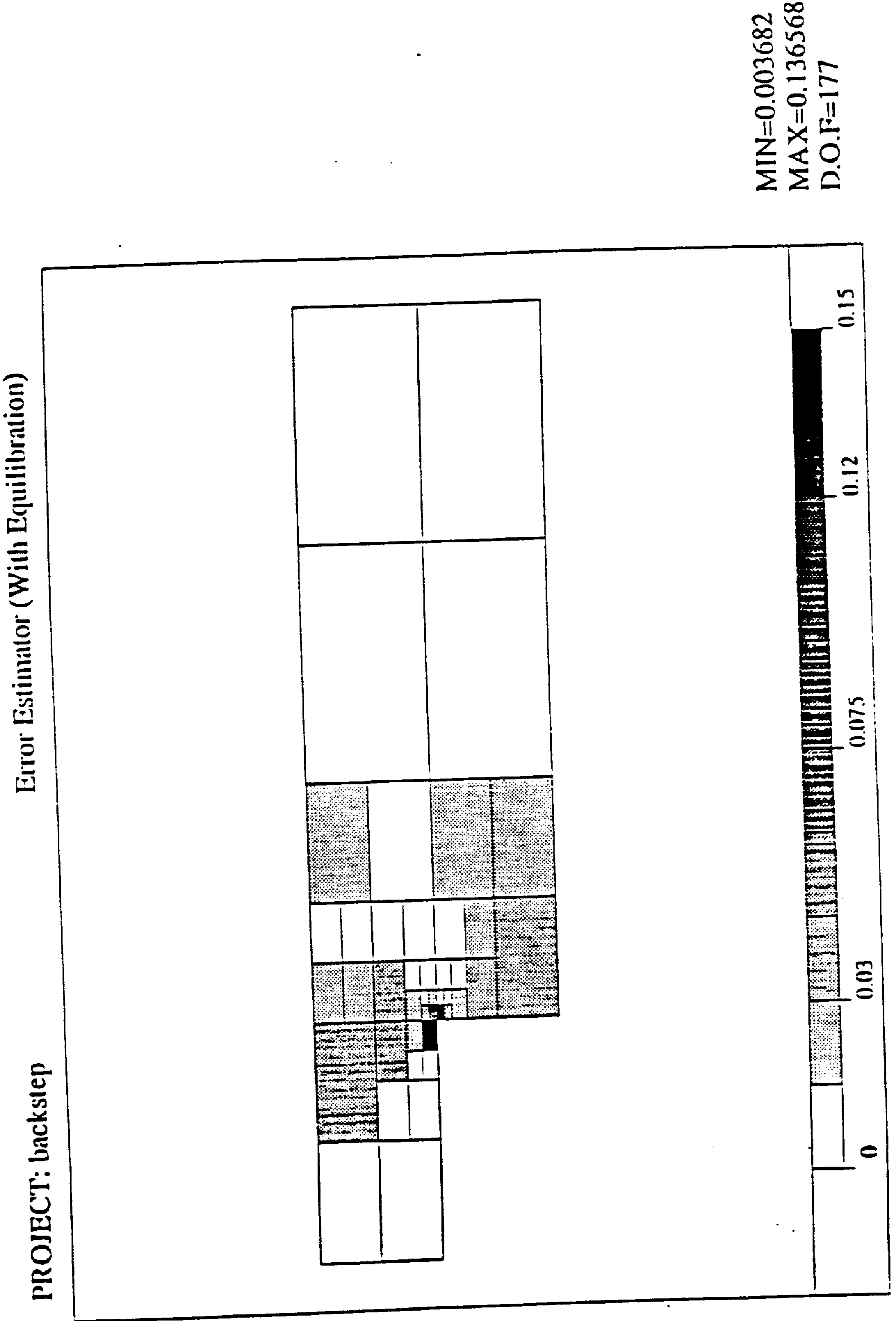


Figure 3.34: Plot of error estimates of solution on third mesh with using flux equilibration.

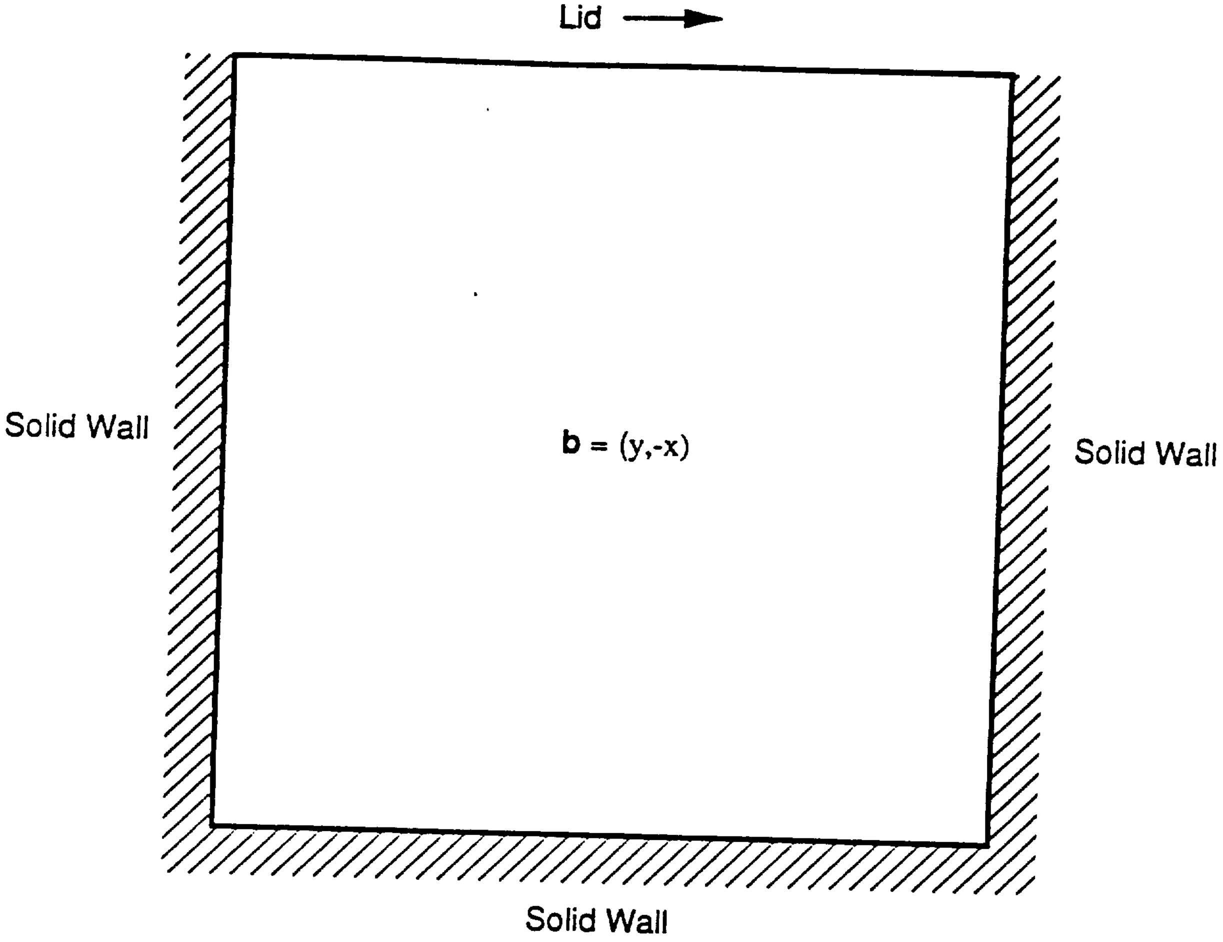


Figure 3.35: Geometry for driven cavity problem.

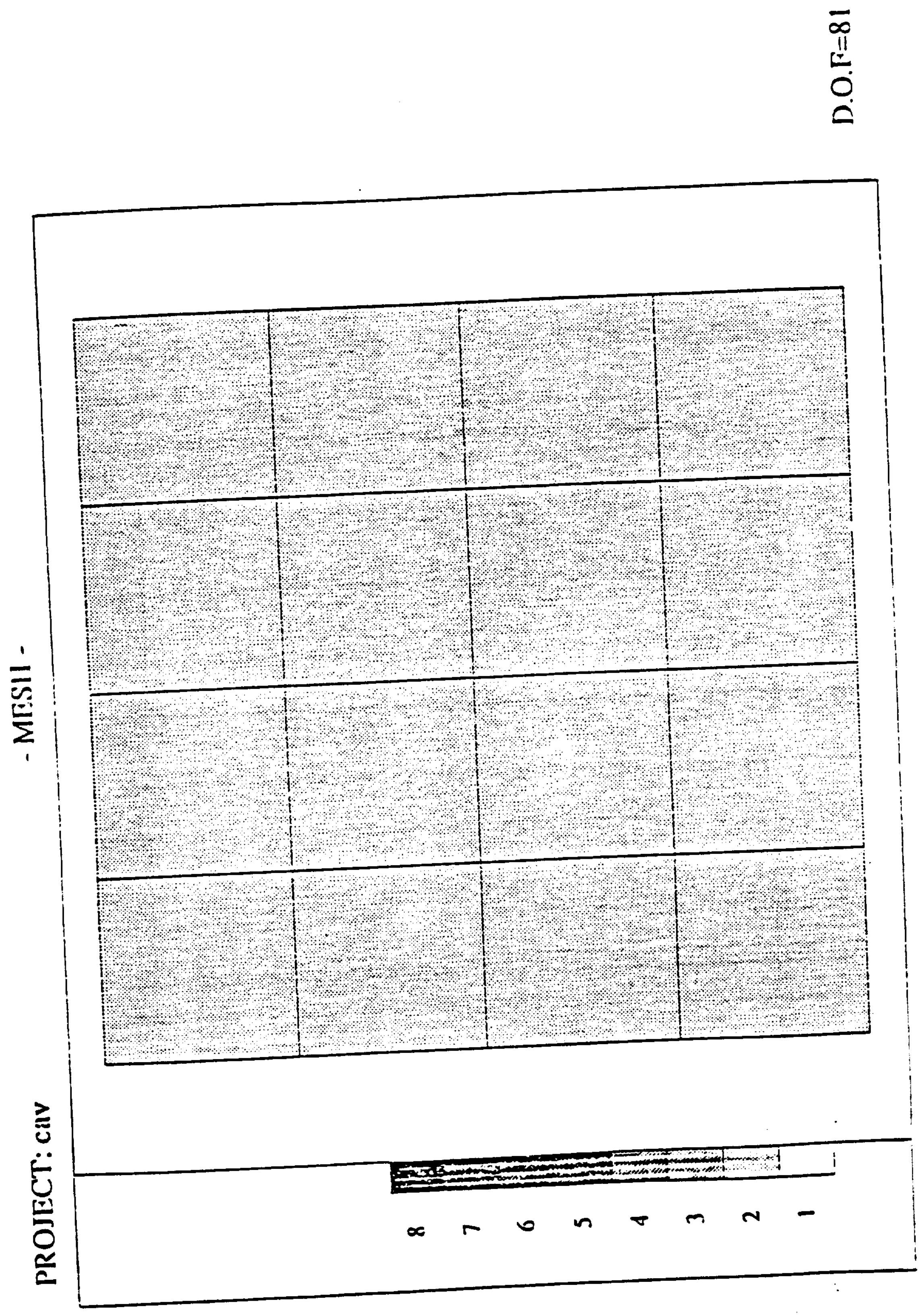


Figure 3.36: First mesh for driven cavity problem.

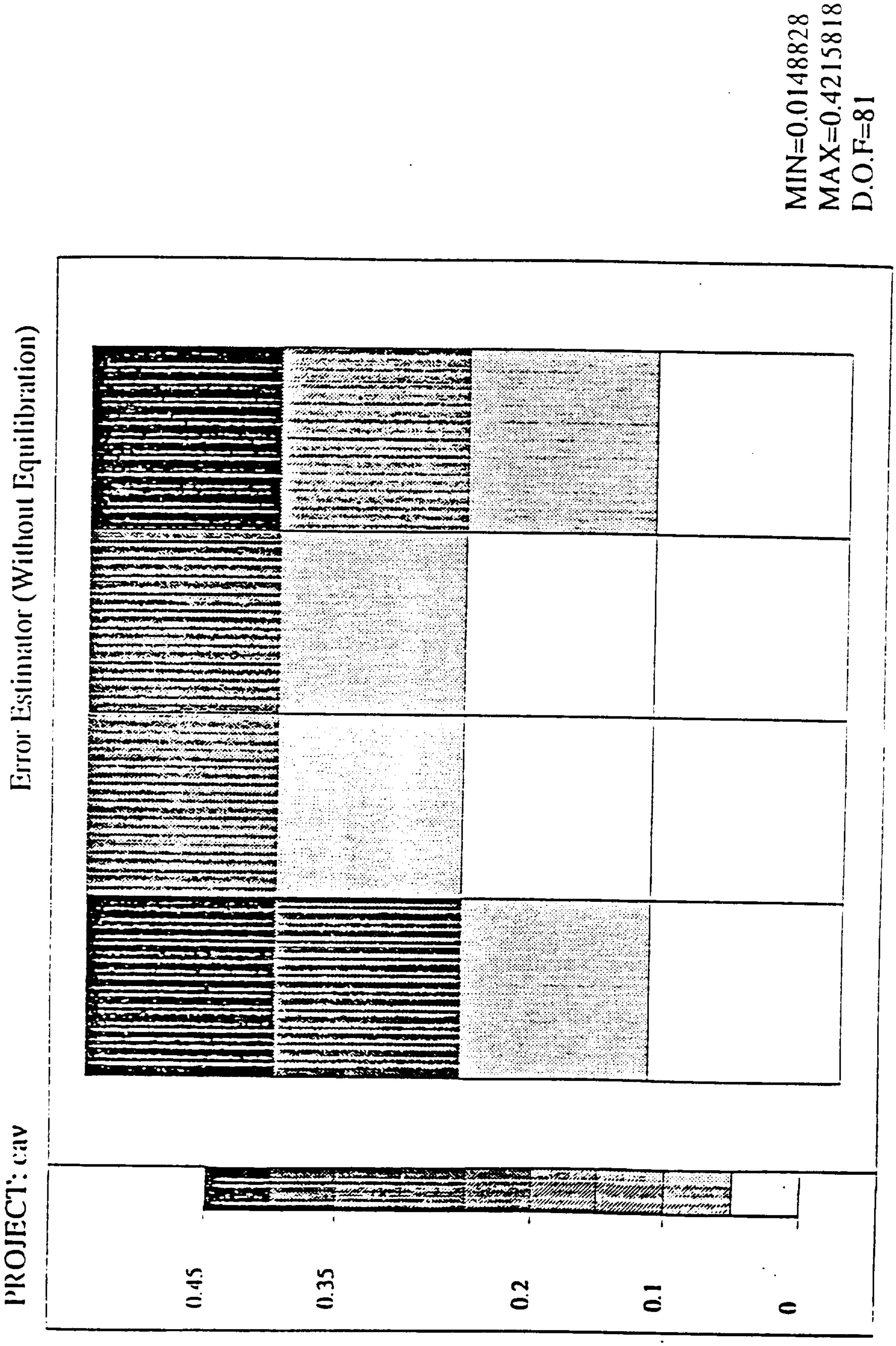


Figure 3.37: Plot of error estimates of solution on first mesh without using flux equilibration.

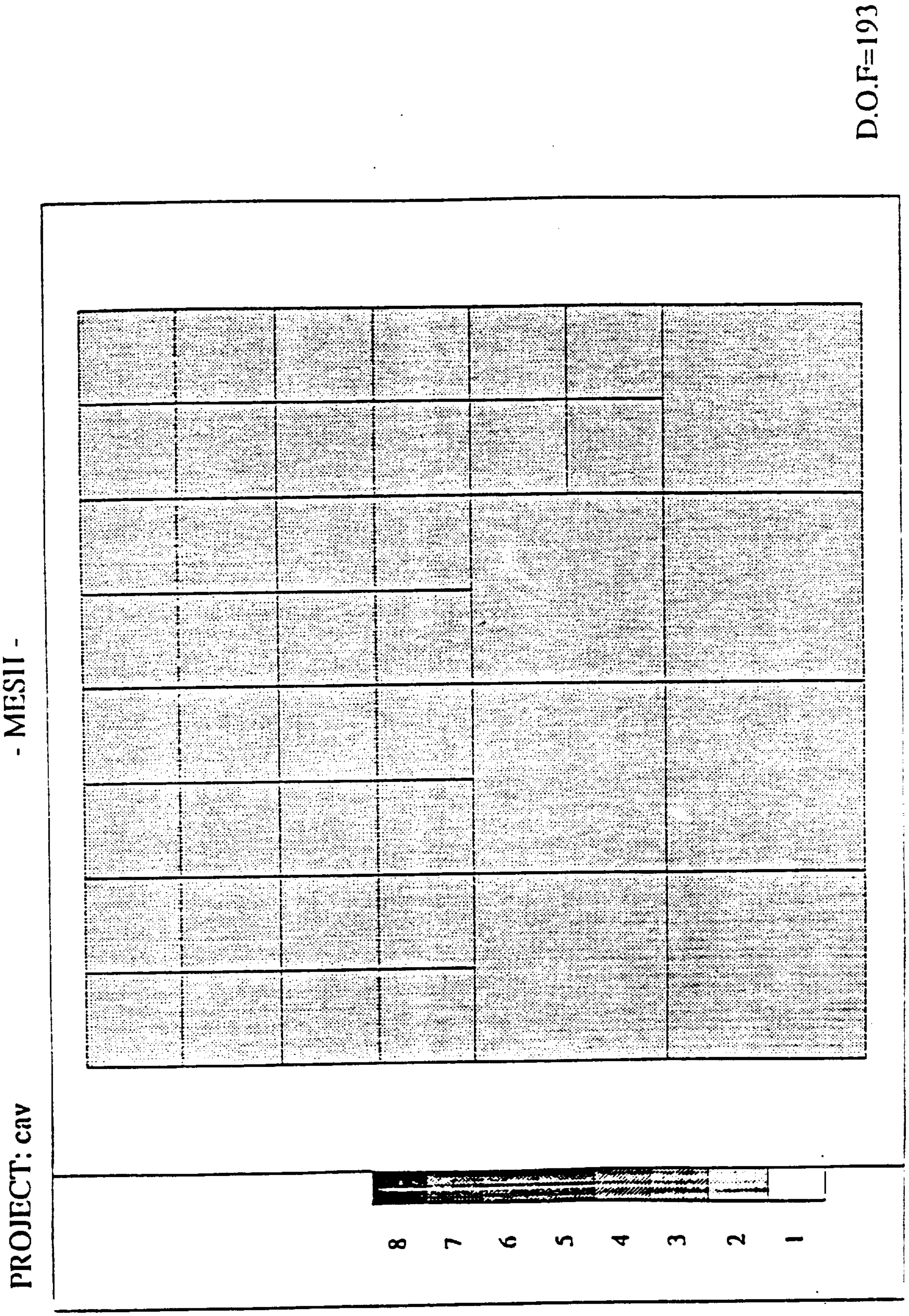


Figure 3.38: Second mesh for driven cavity problem.

C-2

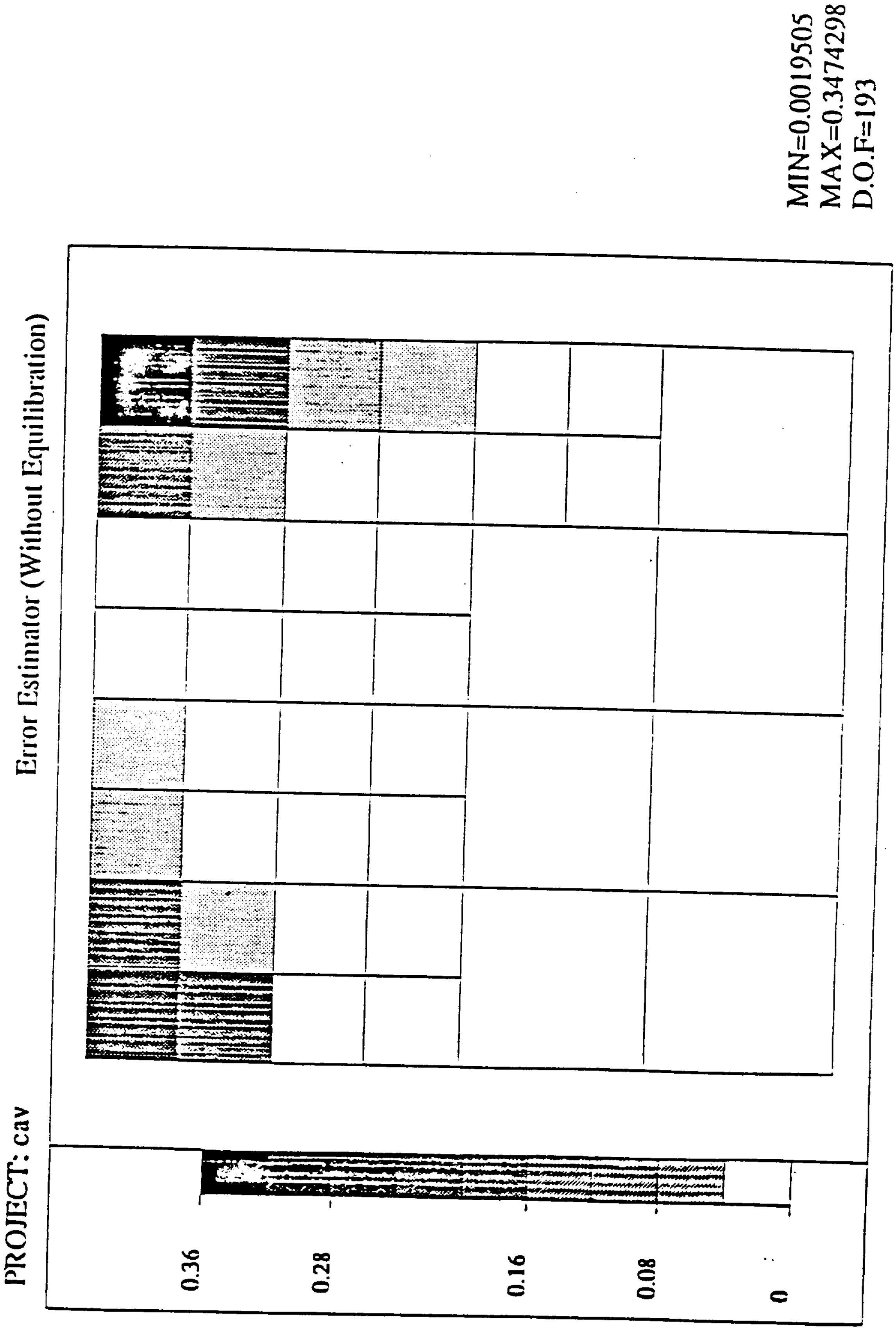


Figure 3.39: Plot of error estimates of solution of second mesh without using flux equilibration.

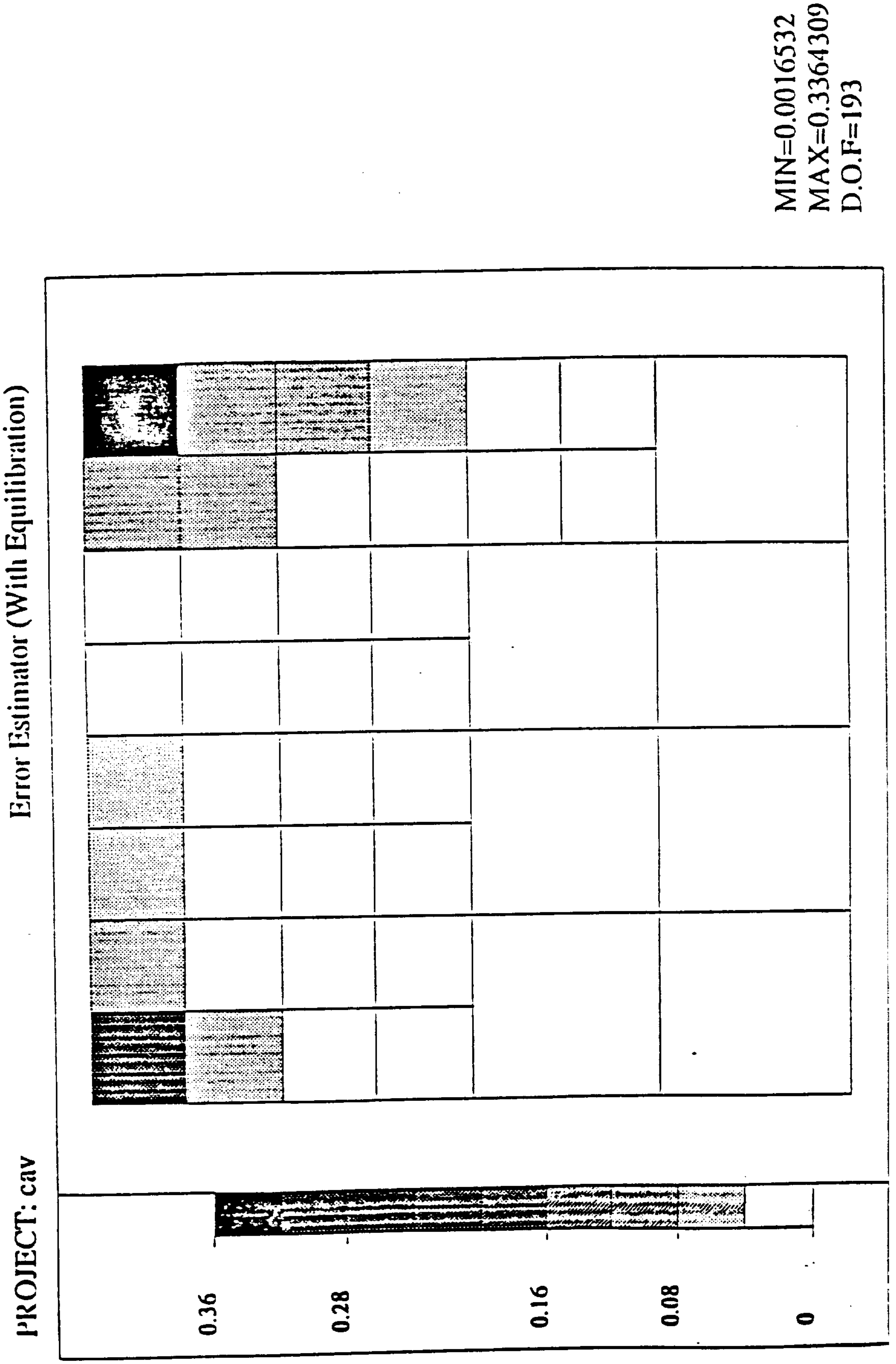


Figure 3.40: Plot of error estimates of solution on second mesh with using flux equilibration.

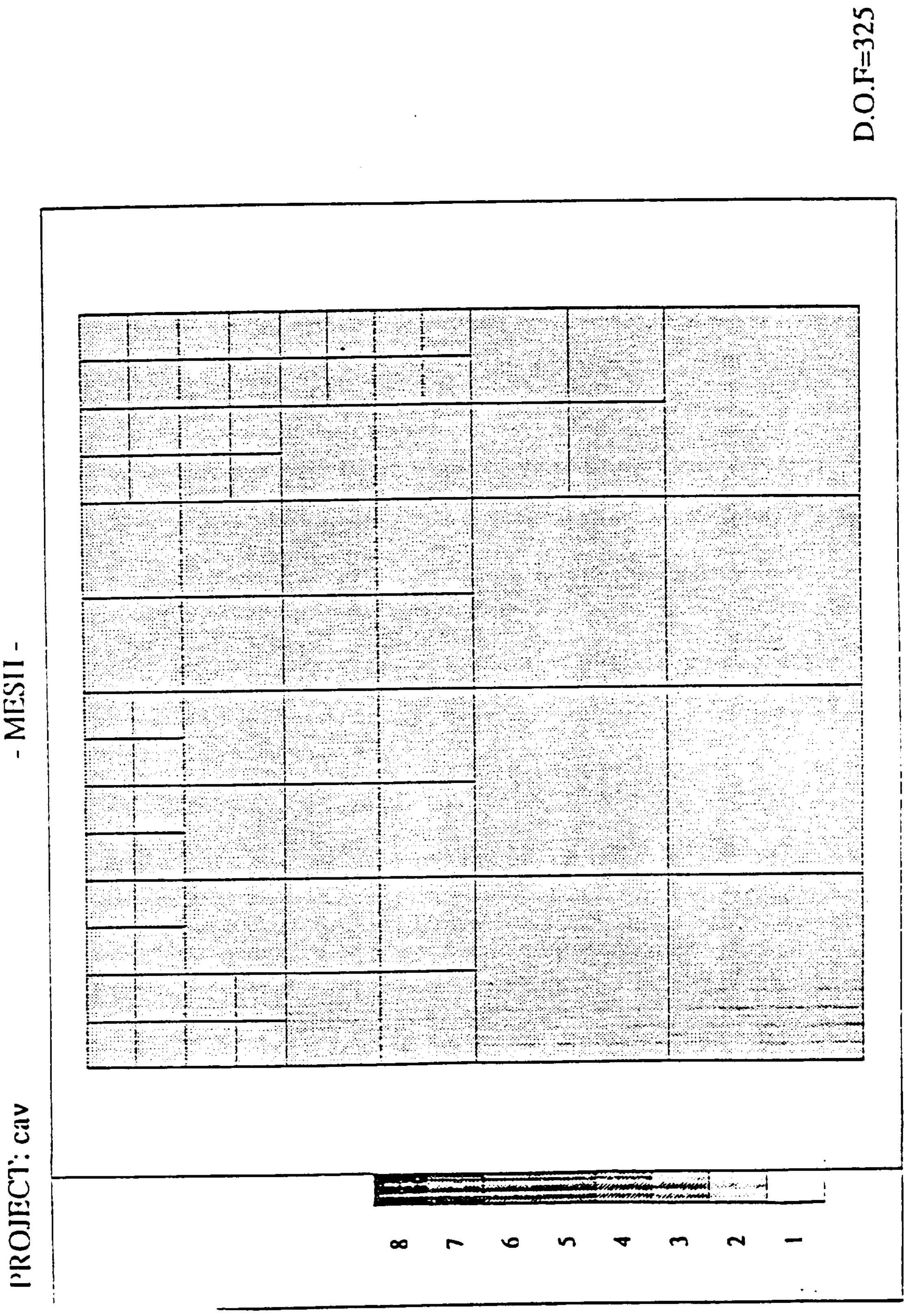


Figure 3.41: Third mesh for driven cavity problem.

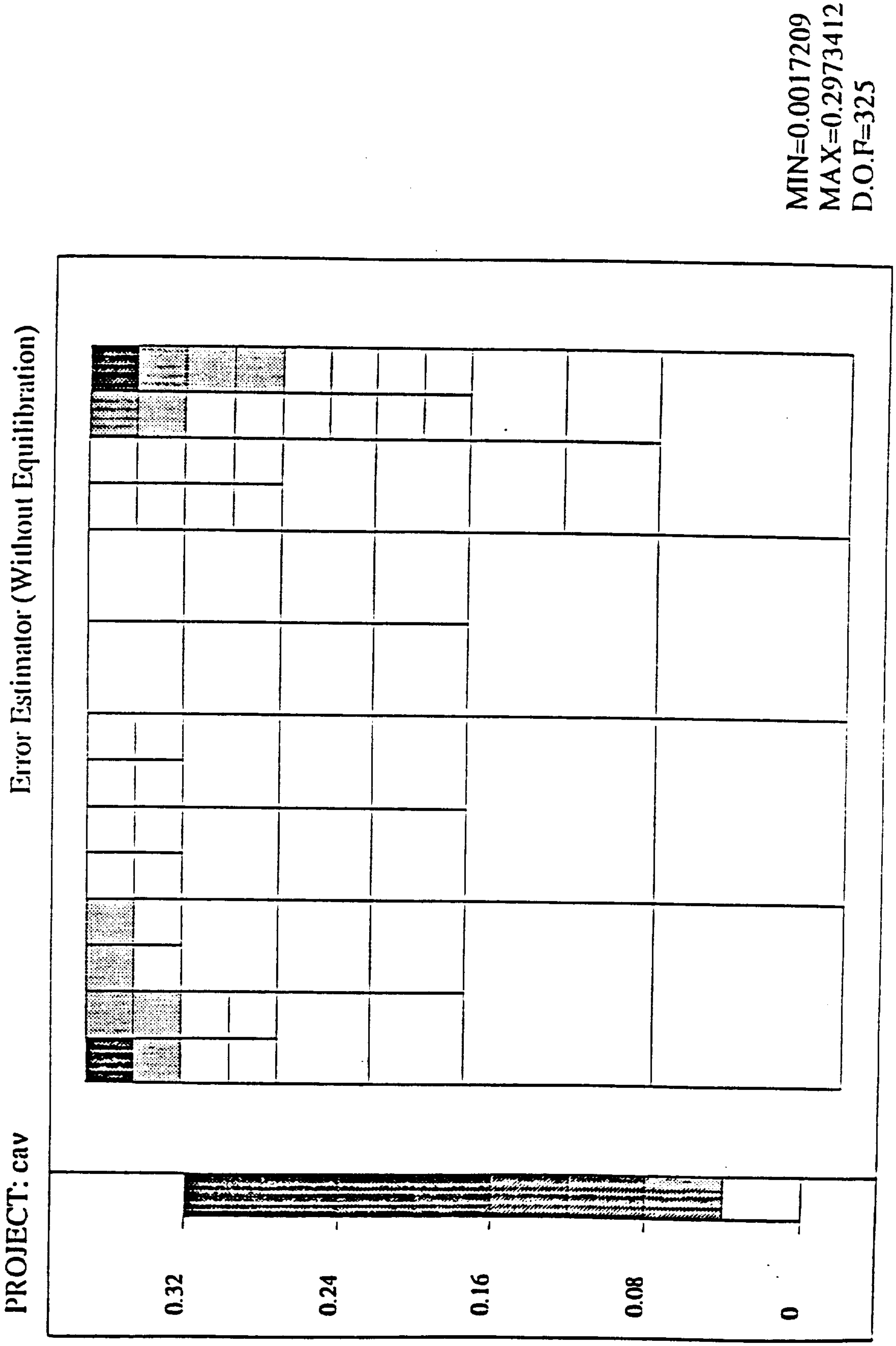


Figure 3.42: Plot of error estimates of solution on third mesh without using flux equilibration.

2

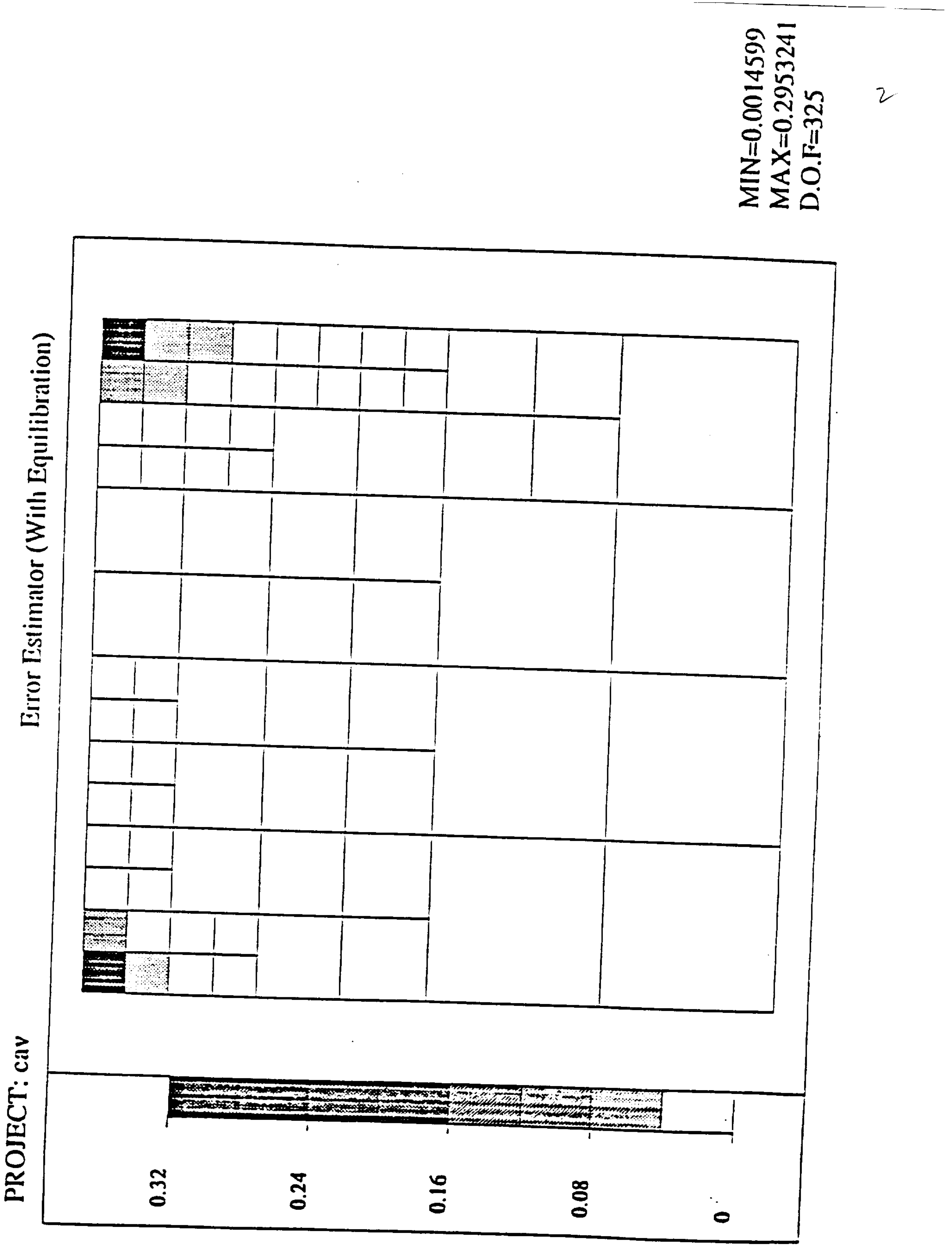


Figure 3.43: Plot of error estimates of solution on third mesh with using flux equilibration.

3.2.5 Residual Error Estimation: A Summary

The residual technique for estimating errors in approximate solutions was discussed in the previous sections. This method seeks to estimate the error $\boldsymbol{e} = \boldsymbol{u} - \boldsymbol{u}_h$ in an approximate solution $\boldsymbol{u}_h$, where the analytic solution $\boldsymbol{u} = (u_1, u_2, \dots u_n)$ satisfies an elliptic system of partial differential equations, which in their general form can be written

$$-\sum_{k,\ell=1}^{3} \frac{\partial}{\partial k}\left(\boldsymbol{A}_{k\ell}\frac{\partial \boldsymbol{u}}{\partial x_\ell}\right) + \sum_{\ell=1}^{3} \overline{\boldsymbol{B}}_\ell \frac{\partial \boldsymbol{u}}{\partial x_\ell} + \boldsymbol{C}\boldsymbol{u} = \boldsymbol{f} \tag{158}$$

subject to the generalized boundary conditions

$$\overline{\boldsymbol{B}}_n \frac{\partial \boldsymbol{u}}{\partial n} + \overline{\boldsymbol{B}} + \frac{\partial \boldsymbol{u}}{\partial t} + \overline{\boldsymbol{B}}_s \frac{\partial \boldsymbol{u}}{\partial s} + \boldsymbol{C}_b\boldsymbol{u} = \boldsymbol{f}_b \tag{159}$$

where the tensors $\boldsymbol{A}_{k\ell}, \overline{\boldsymbol{B}}_\ell, \boldsymbol{C}, \overline{\boldsymbol{B}}_n, \overline{\boldsymbol{B}}_t, \overline{\boldsymbol{B}}_s, \boldsymbol{C}_b, \boldsymbol{f}$ and $\boldsymbol{f}_b$ are known functions of $\boldsymbol{x} = (x, y, z)$ and possibly $\boldsymbol{u}$, and depend upon the particular problem being solved.

After multiplying by a vector of test functions $\boldsymbol{v}$, integrating over the domain (or boundary), and integrating some of the terms by parts, the combined governing equations and boundary conditions can be written in their variational form as:

$$B(\boldsymbol{u}, \boldsymbol{v}) = F(\boldsymbol{v}) \qquad \forall v \epsilon \boldsymbol{H}^1(\Omega) = \underbrace{H^1(\Omega) \times \dots \times H^1(\Omega)}_{\text{n times}} \tag{160}$$

where:

$$\begin{aligned} B(\boldsymbol{u}, \boldsymbol{v}) &= \int_\Omega \left[\sum_{k,\ell=1}^{3} \frac{\partial \boldsymbol{v}^T}{\partial x_k}\boldsymbol{A}_{k\ell}\frac{\partial \boldsymbol{u}}{\partial x_\ell} + \sum_{\ell=1}^{3} \boldsymbol{v}^T\boldsymbol{B}_\ell\frac{\partial \boldsymbol{u}}{\partial x_\ell} + \sum_{k=1}^{3} \frac{\partial \boldsymbol{v}^T}{\partial x_k}\boldsymbol{D}_k\boldsymbol{u} + \boldsymbol{v}^T\boldsymbol{C}\boldsymbol{u}\right] dx \quad (161) \\ &+ \int_{\partial\Omega} \left[\boldsymbol{v}^T\boldsymbol{B}_n\frac{\partial \boldsymbol{u}}{\partial n} + \boldsymbol{v}^T\boldsymbol{B}_t\frac{\partial \boldsymbol{u}}{\partial t} + \boldsymbol{v}^T\boldsymbol{B}_s\frac{\partial \boldsymbol{u}}{\partial s}\right. \\ &+ \left.\frac{\partial \boldsymbol{v}^T}{\partial n}\boldsymbol{D}_n\boldsymbol{u} + \frac{\partial \boldsymbol{v}^T}{\partial t}\boldsymbol{D}t\boldsymbol{u} + \frac{\partial \boldsymbol{v}^T}{\partial s}\boldsymbol{D}_s\boldsymbol{u} + \boldsymbol{C}_b\boldsymbol{u}\right] ds \end{aligned}$$

and

$$\begin{aligned} F(\boldsymbol{v}) &= \int_\Omega \left(\boldsymbol{v}^T\boldsymbol{f} + \sum_{k=1}^{3} \frac{\partial \boldsymbol{v}^T}{\partial x_k}\boldsymbol{g}_k\right) dx \\ &+ \int_{\partial\Omega} \left[\boldsymbol{v}^T\boldsymbol{f}_b + \frac{\partial \boldsymbol{v}^T}{\partial n}\boldsymbol{g}_n + \frac{\partial \boldsymbol{v}^T}{\partial t}\boldsymbol{g}_t + \frac{\partial \boldsymbol{v}^T}{\partial s}\boldsymbol{g}s\right] ds \end{aligned} \tag{162}$$

The corresponding approximate problem used to estimate the error in a given cell or element is formulated:

$$\begin{aligned}
&\text{Find } \boldsymbol{\phi}_h \epsilon \mathcal{M}_h\ (\Omega_k) \text{ such that} && (163)\\
&\overline{B}_k\ (\boldsymbol{\phi}_h, \boldsymbol{w}) - \int_{\partial\Omega_k \backslash \partial\Omega} \sum_{\ell=1}^{3} \boldsymbol{w}^T \boldsymbol{D}_\ell n_\ell \boldsymbol{\phi}\, ds \\
&= F_k\ (\boldsymbol{w}) - B_k\ (\boldsymbol{u}_h, \boldsymbol{w}) - \int_{\partial\Omega_k \backslash \partial\Omega} \left[\sum_{\ell=1}^{3} \boldsymbol{w}^T \boldsymbol{g}_\ell n_\ell - \sum_{\ell=1}^{3} \boldsymbol{w}^T \boldsymbol{D}_\ell n_\ell \boldsymbol{u}_h \right] ds \\
&+ \sum_{\Omega_\ell} \int_{\partial\Omega_\ell \cap\ (\partial\Omega_k \backslash \partial\Omega)} \sum_{i,j=1}^{3} \boldsymbol{\alpha}_\ell \boldsymbol{w}^T \boldsymbol{A}_{ij} \frac{\partial \boldsymbol{u}_h}{\partial x_j} n_i ds \qquad \forall \boldsymbol{w} \epsilon \mathcal{M}_h\ (\Omega_k)
\end{aligned}$$

where B_k and F_k are equations 161 and 162 integrated over Ω_k (instead of Ω) and $\partial\Omega_k \cap \partial\Omega$ (instead of $\partial\Omega$). $\overline{B}_k$ is the same as B_k if B_k is linear, otherwise, $\overline{B}_k$ is an appropriate linearization of B_k. The last term represents a weighted average of the element's and its neighbors' boundary fluxes. Several possible choices exist for the weighting coefficients, $\boldsymbol{\alpha}_\ell$:

- For no average: $\boldsymbol{\alpha}_\ell = (\alpha_\ell, \alpha_\ell, \ldots \alpha_\ell); \alpha_\ell = \delta_{\ell k}$
- For a straight average: $\boldsymbol{\alpha}_\ell = (\alpha_\ell, \alpha_\ell, \ldots \alpha_\ell); \alpha_\ell = \frac{1}{2}$
- For a balanced average: $\boldsymbol{\alpha}_\ell = (\alpha_{\ell 1}, \alpha_{\ell 2}, \ldots \alpha_{\ell n})$; each $\alpha_{\ell i}$ is a piecewise linear function of s obtained by a flux-balancing procedure (see the previous sections.)

The approximate problem represented by equation 163 is solved over a space of polynomial "bubble" functions, which is constructed by first taking the approximation space containing the solution $\boldsymbol{u}_h \epsilon \boldsymbol{H}_{h,p}$ and adding polynomial shape functions until some new complete spectral order is spanned. These added shape functions form the basis for functions in the space $\mathcal{M}_h$. Thus, if the approximate solution consists of (piecewise) polynomials of order p with a corresponding space $\boldsymbol{H}_{h,p}$ and if the enriched space contains polynomials of order $p+1$ (denoted $\boldsymbol{H}_{h,p+1}$), then the three spaces are related according to $\boldsymbol{H}_{h,p+1} = \boldsymbol{H}_{h,p} \oplus \mathcal{M}_h$. For example, if the approximate solution on a brick element is trilinear (8 linear shape functions) and the enriched space is triquadratic (8 linear, 1 fully quadratic and 18 partially quadratic/partially linear shape functions), then the bubble function space $\mathcal{M}_h$ is spanned by the 19 additional quadratic shape functions.

Once the error indicator functions ϕ have been calculated, the error estimate is computed as a norm of ϕ. Namely,

$$\epsilon_k = |||\phi||| = \sqrt{\overline{\overline{B}}(\phi, \phi)} \qquad (164)$$

where the precise form of $\overline{\overline{B}}(\cdot,\cdot)$ depends primarily upon the type of norm desired. For energy type norms, $\overline{\overline{B}}(\cdot,\cdot)$ may be very similar to other forms, $\overline{B}(\cdot,\cdot)$ and $B(\cdot,\cdot)$. In fact, for symmetric linear problems such as heat transfer or linear elasticity problems, all three forms are identical.

In summary, the process of computing residual error estimates for a given mesh consists of four steps for each element.

1. Compute the contributions of the domain integrals of equations 161 and 162 to the left and right hand sides of the linear algebra problem (henceforth denoted as "stiffness matrix" and "load vector").
2. Compute the contributions of the various boundary integrals of equations 161, 162, and 163 to the stiffness matrix and load vector.
3. Solve the linear algebra problem to obtain the error indicators.
4. Evaluate the norms of the indicators to get the final error estimates.
5. Evaluate the mesh-wise norm of the solution to obtain a scaling factor.
6. Normalize each error estimate by the scaling factor.

3.3 Solution Enhancement

This section describes the work done in the area of solution enhancement techniques. It includes summaries of the results of a literature survey and research performed on this topic.

3.3.1 An Overview of Postprocessing Techniques

One paradoxical feature of the finite element method is that it often delivers a rather poor approximation to the quantities of the most interest. For example, when solving an elasticity problem, we are primarily concerned with the stresses. However, the finite element method provides an approximation of the displacement. Obviously, by a process of differentiation one can use the approximation to the displacement to obtain an approximation to the stress. Unfortunately, this approach often results in a very poor approximation to the stress. In particular, the stresses will be discontinuous across the interelement boundaries.

This characteristic of the finite element method has been extensively studied and consequently various techniques are often used to enhance stress approximations. Such techniques are usually referred to as *postprocessing techniques.*

In this section we shall be concerned with attempting to find suitable postprocessing techniques for use with various classes of approximations including finite difference, finite volume, and finite element approximations. As we shall see, many of the existing techniques are applied to very specific types of finite element approximation schemes and in some cases are strongly problem-dependent. It is not only that the schemes have not been analyzed mathematically—they just fail completely when applied to other situations.

The remainder of this section is organized as follows. First, we give a survey of the most popular and successful postprocessing techniques in use today. Second, we propose several techniques which hold the greatest promise in the context of general approximations. Throughout this and the following sections we shall illustrate the basic concepts by using an example of finite element discretization of a simple model problem:

Find u such that

$$\begin{aligned} -\Delta u + u &= f \quad \text{in} \quad \Omega \\ u &= 0 \quad \text{on} \quad \Gamma_D \\ \frac{\partial u}{\partial n} &= g \quad \text{on} \quad \Gamma_N \end{aligned} \tag{165}$$

where Ω is a domain in $\mathbb{R}^2$ with boundary $\partial\Omega = \overline{\Gamma_n \cup \Gamma_D}$ and where $\Gamma_n \cap \Gamma_D$ is empty. We shall let $\hat{u}$ denote a standard finite element approximation to u. The actual details of the finite element approximation scheme will be made clear in each case, and will be assumed to be general h-p finite element approximation otherwise.

Superconvergence Phenomenon

Perhaps the most natural types of postprocessing schemes have their roots in the phenomenon known as *superconvergence*. Indeed, this phenomenon is discussed even in the early work of Strang and Fix [28]. Probably the work of Barlow [29] was one of the first to make use of the idea that at certain "exceptional" or so-called "stress" points, the accuracy of the direct approximation $\nabla\hat{u}$ to the gradient is significantly higher than the accuracy at other points. In order to give a concrete illustration we consider the case of our model problem in one dimension.

$$\begin{gathered} -u'' + u = f \text{ on } [0,1] \\ u(0) = u(1) = 0 \end{gathered} \tag{166}$$

We consider a uniform mesh of size $h > 0$ and uniform polynomial degree p on each element, see Fig. 3.44.

Let $\hat{u}$ denote the finite element approximation. In order to postprocess the gradient $\hat{u}'$ on element Ω_k we evaluate $\hat{u}'$ at the mapped Gauss-Legendre points. That is to say, we let $\xi_1, \xi_2, \ldots, \xi_p$ denote the zeros of the p-th degree Legendre polynomial.

These lie on the interval $[-1,1]$ which we map to element Ω_k using the transformation

$$F_k\colon[-1,1] \to \Omega_k,\ F_k(\xi) = \frac{1}{2}a(1-\xi) + \frac{1}{2}b(1+\xi) \tag{167}$$

where $\Omega_k = [a,b]$. We denote $x_j^{(k)} = F_k(\xi_j)$. Then the average error in the approximation to the true gradient at these mapped Gauss points decreases at a rate of an order of h greater than the approximation elsewhere. Expressed precisely, the result is

$$\left\{\sum_{k=1}^{N}\sum_{j=1}^{P} h\left|u'\left[x_j^{(k)}\right] - \hat{u}'\left[x_j^{(k)}\right]\right|^2\right\}^{\frac{1}{2}} \leq Ch^{p+1} \tag{168}$$

whereas at other points the rate of convergence is $O(h^p)$. In short, *$\hat{u}'$ is an order of h more accurate at the mapped Gauss points.*

Of course having identified the points at which superconvergence occurs, we can obtain a super accurate solution at any other point in the domain by interpolating through these special points. Figure 3.45 shows the scheme in the case $p = 1$.

If we let $G(\hat{u}')$ denote the interpolated gradient, then it has been shown in [15] under certain assumptions:

$$\|u' - G(\hat{u}')\|_{L_2(\Omega)} \leq Ch\,\|u' - \hat{u}'\|_{L_2(\Omega)} \tag{169}$$

and that

$$\|u'' - G'(\hat{u}')\|_{L_2(\Omega)} \leq Ch^p \tag{170}$$

This one-dimensional example illustrates the key ideas used by this technique. Unfortunately, their extension into higher dimensions is not straightforward and will be discussed for several two-dimensional element types in the next sections.

Linear Triangles

Consider the case of an approximation using polynomial degree 1 shape functions on triangular elements. It is well known [30,31] that there are no points at which the superconvergence is observed. Indeed, it was thought that it was impossible to obtain superconvergence, until the work of Levine [30] revealed that a more subtle approach based on averaging gradients between neighboring elements can be employed. In order to explain Levine's idea, we imagine a pair of triangular elements in a large mesh, see Fig. 3.46.

Levine showed that a superconvergent gradient can be obtained at the midpoint of the interelement boundary as follows: define

$$\boldsymbol{G}(\nabla\hat{u})(\boldsymbol{x}*) = \frac{1}{2}\left\{\nabla\hat{u}\,|_{\Omega_1}(\boldsymbol{x}*) + \nabla\hat{u}|_{\Omega_2}(\boldsymbol{x}*)\right\} \tag{171}$$

where $\nabla\hat{u}|_{\Omega_1}$ denotes the finite element approximation to the gradient on element Ω_1. It turns out that $G(\nabla\hat{u})(\boldsymbol{x}*)$ is a superconvergent approximation. The assumptions needed to show this result mathematically are that u is smooth (i.e., $u \in H^3(\Omega)$) and that *the pair of triangles form a parallelogram.*

Translated into a statement regarding the whole mesh, the second assumption leads to the famous "six triangle property." Namely, that each interior node in the mesh must be surrounded by precisely six triangles. Examples of meshes which satisfy and fail this property are shown in Fig. 3.47.

One is severely tempted to dismiss the "six triangle" property as an artificial mathematical requirement which can be safely ignored in practice. Unfortunately, *it is necessary for superconvergence.* Without this property one does not obtain superconvergence.

Quadratic Basis Functions on Triangles

The situation here is very similar to that for linear triangles. Namely, one is forced to average between pairs of triangles to obtain superconvergence. Correspondingly, the six triangle property must also hold. Figure 3.48 shows the stress points for quadratic elements.

It has been shown [32] that the two Gauss points along the interelement edges can be used to obtain superconvergence.

$$G(\nabla\hat{u})(\boldsymbol{x}*_i) = \frac{1}{2}\left\{\nabla\hat{u}|_{\Omega_1}(\boldsymbol{x}*_i) + \nabla\hat{u}|_{\Omega_2}(\boldsymbol{x}*_i)\right\} \qquad i = 1,2 \tag{172}$$

For the result to hold, one must assume that u is very smooth (i.e., $u \in H^4(\Omega)$). Once again, without the smoothness assumption and the six triangle property, the superconvergence is destroyed or severely degraded.

High–Order Basis Functions on Linear Triangles

For the case of cubic, or higher order basis functions on triangles, one is tempted to conjecture that the Gauss points on the interelement edge are the appropriate points where superconvergence occurs. Unfortunately this does not appear to be the case. To date, no one has been able to demonstrate superconvergence for cubic or higher order triangular elements. The general opinion in the "superconvergence fraternity" is that it does not occur and that it is impossible.

Quadrilateral Elements

While this superconvergence technique seems to be of limited usefulness for triangular elements, the same is not true for quadrilateral elements. The fundamental results for these types of elements were shown first by Lesaint and Zlamal [33]. Unlike triangles, superconvergence occurs for all orders of quadrilateral elements. However, as with triangular elements, the solution u must be very regular ($u \in H^{p+2}(\Omega)$) and also the mesh must be very regular. Essentially the elements must be parallelograms and the overall partitioning must be topologically equivalent to a mesh of the form shown in Fig. 3.49. That is, there must be four quadrilaterals around each node implying a structured mesh.

Given a single element Ω_k from the mesh and assuming that $\hat{u}$ is a piecewise polynomial of degree p, the points at which superconvergence occurs are at the images of the pairs of Gauss-Legendre points. That is, if $F_k: Q \to \Omega_k$ where $Q = [-1,1] \times [-1,1]$ is the standard master element, then the superconvergent stress points are at $F_k(\xi_i, \xi_j)$ $1 \leq i,j \leq p$ where $\{\xi_i\}$ are the zeros of the p-th degree Legendre polynomial.

Summary of Superconvergence Phenomenon

The primary advantages of using superconvergence as a basis for postprocessing are its ease of implementation and that it is extremely inexpensive. Also, just because the technique is inexpensive does not imply that it offers a limited improvement to the solution. When the assumptions are satisfied, the results are sometimes startling. However, the main disadvantages are the regularity requirements on the mesh and the solution.

Projection Methods

While the superconvergence or averaging approach evolved largely from the mathematical community, an alternative technique has evolved from the engineering community. The basic idea behind the method can be explained as follows:

We require a continuous approximation to the gradient, but $\nabla\hat{u}$ is discontinuous across interelement boundaries. Let $\varphi_1, \varphi_2, \ldots, \varphi_N$ denote the shape functions used to obtain $\hat{u}$. Then we write our approximation $\boldsymbol{G}(\hat{u})$ to the gradient in the form

$$\boldsymbol{G}(\hat{u})(\boldsymbol{x}) = \left(\sum_{j=1}^{N} \alpha_j \varphi_j(\boldsymbol{x}), \sum_{j=1}^{N} \beta_j \varphi_j(\boldsymbol{x}) \right) \tag{173}$$

where $\alpha_1, \alpha_2, \ldots, \alpha_N$ and $\beta_1, \beta_2, \ldots, \beta_N$ are constants to be determined. Since $\boldsymbol{G}$ will automatically be a continuous approximation to the gradient the question is how to determine these multipliers. A projection technique is used to determine

$$J(\alpha_1, \ldots, \alpha_N; \beta_1, \ldots, \beta_N) = \int_\Omega \{\boldsymbol{G}(\hat{u})(\boldsymbol{x}) - \nabla\hat{u}(\boldsymbol{x})\}^2 \, d\boldsymbol{x} \tag{174}$$

Writing down the conditions for a minimum point of J we obtain the equations

$$M\boldsymbol{\alpha} = \boldsymbol{g}_x \tag{175}$$

and

$$M\boldsymbol{\beta} = \boldsymbol{g}_y \tag{176}$$

where M is the $N \times N$ matrix with elements

$$M_{ij} = \int_\Omega \varphi_i(\boldsymbol{x})\varphi_j(\boldsymbol{x}) d\boldsymbol{x} \tag{177}$$

and $\boldsymbol{g}_x, \boldsymbol{g}_y$ are the N-vectors with elements

$$
\left.\begin{aligned}
[g_x]_i &= \int_\Omega \varphi_i(\boldsymbol{x}) \frac{\partial \hat{u}}{\partial x}\, d\boldsymbol{x} \\
[g_y]_i &= \int_\Omega \varphi_i(\boldsymbol{x}) \frac{\partial \hat{u}}{\partial y}\, d\boldsymbol{x}
\end{aligned}\right. \qquad (178)
$$

The matrix M is guaranteed to be non-singular and hence $\boldsymbol{\alpha}$ nd $\boldsymbol{\beta}$ are uniquely determined from these equations. This approach is identical to the approach known as the "Conjugate Flux Method" of [34]. In the above notation and context, the method was described by Hinton and Campbell [35]. Surprisingly enough, there is little or no mathematical analysis to support this method [36]. However, numerical evidence shows that the approach can be extremely effective. A related method was presented and analyzed exhaustively by Bramble and Schatz [37], although they made very stringent requirements on the mesh without which it is not clear how their approach can even be implemented.

More recently, this approach has gained in popularity owing to its application to error estimation using the Zienkiewicz-Zhu method [16,38].

Analysis presented in [36,37] along with extensive numerical experience suggests that the method does indeed yield superconvergence but not globally. Namely, near the boundary of the domain, the method does not only fail to enhance the accuracy, but even degrades it. Unfortunately, it is usually near the boundary where we are the most interested in the gradients or stresses, since this is where singularities occur in applications such as elasticity.

This phenomena is illustrated in the context of error estimation in Figures 3.50 - 3.54 for a three-dimensional linear elasticity problem. This problem corresponds to the pure bending of a prismatic beam. Figure 3.50 shows a sketch of the beam and its loading configuration along with a picture of the deformed shape of the body. Since the analytic solution to this problem is a second-order polynomial, only meshes of linear elements were used. Figures 3.51 - 3.54 show plots of the error estimates and analytic errors measured in the energy norm for a sequence of meshes. The energy corresponding to the analytic solution is 223.6068. Based upon these figures, we see that this technique provides reasonable estimates of the global error but poor indications of how that error is distributed. This is especially true in elements near the ends of the beam.

A further disadvantage of the method is the vast expense entailed in solving the matrix equations for $\boldsymbol{\alpha}$ and $\boldsymbol{\beta}$. In fact, for two-dimensional problems the cost of postprocessing is roughly twice the cost of obtaining $\hat{u}$ itself.

Cantin-Loubignac Iteration

Let us consider the linear elasticity problem:

$$
\left.\begin{aligned}
-S^t D S \boldsymbol{u} &= \boldsymbol{f} \text{ in } \Omega \\
\boldsymbol{u} &= \boldsymbol{0} \text{ on } \Gamma_D \\
H\boldsymbol{\sigma} &= \boldsymbol{g} \text{ on } \Gamma_N
\end{aligned}\right\} \qquad (179)
$$

where $\boldsymbol{u}$ is the displacement, $\boldsymbol{\sigma} = DS\boldsymbol{u}$ is the stress, D is the elasticity matrix, S is the operator

$$S = \begin{bmatrix} \partial x & 0 \\ 0 & \partial y \\ \partial y & \partial x \end{bmatrix} \tag{180}$$

and H is formed from the unit normal vector $\boldsymbol{n} = (n_x, n_y)$ as follows:

$$H = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix} \tag{181}$$

We let $\hat{\boldsymbol{u}}$ be a finite element approximation to $\boldsymbol{u}$. Furthermore, let $\boldsymbol{G}(\hat{\boldsymbol{u}})$ denote a recovered approximation to be the stress obtained using any of the techniques we have just discussed.

The so–called Cantin-Loubignac iteration is a process for further enhancing the accuracy of $\boldsymbol{G}(\hat{\boldsymbol{u}}) \approx \boldsymbol{\sigma}$. The variational form of (179) is to find $\boldsymbol{u}$ such that

$$\boldsymbol{u} \in X : B(\boldsymbol{u}, \boldsymbol{v}) = (\boldsymbol{f}, \boldsymbol{v}) + \int_{\Gamma_N} \boldsymbol{g} \cdot \boldsymbol{v} \, ds \qquad \forall\, \boldsymbol{v} \in X \tag{182}$$

where

$$B(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega (S\boldsymbol{v})^t D(S\boldsymbol{u}) d\boldsymbol{x} \tag{183}$$

$$(\boldsymbol{f}, \boldsymbol{v}) = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, d\boldsymbol{x} \tag{184}$$

and

$$X = \left\{ \boldsymbol{u} \in \left[H^1(\Omega)\right]^2 : \boldsymbol{u} = \boldsymbol{0} \text{ on } \Gamma_D \right\} \tag{185}$$

If our approximation to the stress is exact, then we would have

$$\int_\Omega (S\boldsymbol{v})^t \boldsymbol{G}(\hat{\boldsymbol{u}}) \, d\boldsymbol{x} = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, d\boldsymbol{x} + \int_{\Gamma_N} \boldsymbol{g} \cdot \boldsymbol{v} \, ds \qquad \forall\, \boldsymbol{v} \in X \tag{186}$$

Thus, the lack of accuracy in $\boldsymbol{G}(\hat{\boldsymbol{u}})$ is measured by the quantity

$$(\boldsymbol{r}, \boldsymbol{v}) = \int_\Omega \boldsymbol{f} \cdot \boldsymbol{v} \, d\boldsymbol{x} + \int_{\Gamma_N} \boldsymbol{g} \cdot \boldsymbol{v} \, , d\boldsymbol{s} - \int_\Omega (S\boldsymbol{v})^t \boldsymbol{G}(\hat{\boldsymbol{u}}) \, d\boldsymbol{x} \tag{187}$$

Since $\boldsymbol{G}(\hat{\boldsymbol{u}})$ is continuous (by construction) we may integrate (187) by parts to obtain

$$(\boldsymbol{r},\boldsymbol{v}) = \int_{\Omega} \left\{\boldsymbol{f} - S^t\boldsymbol{G}(\hat{\boldsymbol{u}})\right\} \cdot \boldsymbol{v}\, d\boldsymbol{x} + \int_{\Gamma_D} \{\boldsymbol{g} - H\boldsymbol{G}(\hat{\boldsymbol{u}})\} \cdot \boldsymbol{v}\, ds \quad \forall\, \boldsymbol{v} \in X \tag{188}$$

Notice that every term on the right hand side is computable. Furthermore, we have from (187)

$$(\boldsymbol{r},\boldsymbol{v}) = B(\boldsymbol{u},\boldsymbol{v}) - \int_{\Omega}(S\boldsymbol{v})^t\boldsymbol{G}(\hat{\boldsymbol{u}})\, d\boldsymbol{x} = \int_{\Omega}(S\boldsymbol{v})^t\{\boldsymbol{\sigma} - \boldsymbol{G}(\hat{\boldsymbol{u}})\}\, d\boldsymbol{x} \tag{189}$$

The method is therefore to first solve the approximate correction problem

$$\widehat{\boldsymbol{w}} \in \widehat{X} \quad B(\widehat{\boldsymbol{w}},\widehat{\boldsymbol{v}}) = (\boldsymbol{r},\widehat{\boldsymbol{v}}) \quad \forall\, \widehat{\boldsymbol{v}} \in \widehat{X} \tag{190}$$

by means of a finite element discretization precisely the same as that used to obtain $\hat{\boldsymbol{u}}$. Having calculated $\widehat{\boldsymbol{w}}$, we then set

$$\hat{\boldsymbol{u}}* = \hat{\boldsymbol{u}} + \widehat{\boldsymbol{w}} \in \widehat{X} \tag{191}$$

and compute $\hat{\boldsymbol{\sigma}}* = \boldsymbol{G}(\hat{\boldsymbol{u}}*)$ in the same way as before. It is clear that $\hat{\boldsymbol{\sigma}}* \approx \boldsymbol{\sigma}$. In fact, it is found that $\hat{\boldsymbol{\sigma}}*$ is a *much better approximation* to the true stress than $\boldsymbol{G}(\hat{\boldsymbol{u}})$. Of course, having obtained $\hat{\boldsymbol{\sigma}}*$, the process can be repeated as often as we like. It is found computationally that the sequence of iterates tends toward $\boldsymbol{\sigma}$, and that convergence of the iterates occurs in only a few iterations.

The philosophy behind this approach appears similar to the *Defect Correction Technique* frequently used in numerical analysis. However, there are certain differences making the question of a mathematical analysis somewhat recalcitrant. In spite of this, the method works in practice. The major disadvantage associated with the method is the significant computational expense entailed while its major advantage is its generality.

Extraction Methods

A good overview of these techniques was presented in [39] which is summarized in this section. The concept of extraction formulas was first developed by Babuška and Miller [40,41,42]. They noticed that some properties of solutions u to a given boundary value problem can be characterized by integral identities depending on u which they called extraction formulas, which are similar to Green's formulas for elliptic problems. They proposed to calculate these properties with extraction formulas but with the exact solution u replaced by its FE approximation $\hat{u}$. It turns out that with this method we can "extract" at a given point the value of u, derivative of $u(x)$, its average value, etc., and the results are more accurate than those obtained by direct computation.

Some spectacular examples showing improvement of accuracy due to applying extraction formulas were presented in [40] and [41]. Among them, the evaluation of stress intensity factors seems to be the most interesting (though, in this case, their approach is very closely related to other commonly known methods of linear fracture mechanics, for instance, [43]).

However, in none of the work in [41] was the problem of estimating gradients for two-dimensional problems considered at all, except for the calculation of normal derivatives on the boundary $\partial\Omega$ of the domain of u.

Here we follow the general idea of the method described in [40] and we develop appropriate formulas for extracting $\partial u/\partial x$. We focus our attention on the model elliptic problem:

$$\begin{aligned} -\frac{\partial}{\partial x_i}\left(a(\boldsymbol{x})\frac{\partial u}{\partial x_i}\right) &= f(\boldsymbol{x}) \text{ in } \Omega \\ u &= u_0 \text{ on } \partial\Omega \end{aligned} \tag{192}$$

Integrating twice by parts the product of this equation and any function $(\varphi + \tilde{\varphi})$, we obtain the following identity:

$$\Phi = \int_\Omega f \cdot (\varphi + \tilde{\varphi})\, d\boldsymbol{x} + \int_\Omega \left[-\left(a\tilde{\varphi}_{,i}\right)_{,i} u\right] d\boldsymbol{x} + \int_\Omega \left[(a - a_0)\left(\varphi + \tilde{\varphi}\right)_{,i}\right]_{,i} u d\boldsymbol{x} \tag{193}$$

where

$$\begin{aligned} \Phi &= 0 \text{ if } (\varphi + \tilde{\varphi}) \text{ is smooth, at least a } C^1 \text{ function,} \\ \Phi &= u(\boldsymbol{y}) \text{ if } \varphi(\boldsymbol{x}, \boldsymbol{y}) = \frac{-1}{2\pi a_0} \log|\boldsymbol{x} = \boldsymbol{y}|, \\ \Phi &= \frac{\partial u}{\partial x_1}(\boldsymbol{y}) \text{ if } \varphi(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{2\pi a_0}\frac{\cos\alpha}{|\boldsymbol{x} - \boldsymbol{y}|},\ \cos\alpha = \frac{x_1 - y_1}{|\boldsymbol{x} - \boldsymbol{y}|}, \\ a_0 &= a(\boldsymbol{y}), \\ \tilde{\varphi} &= \text{any smooth function } (C^1) \text{ satisfying:} \end{aligned}$$

$$\begin{aligned} (\varphi + \tilde{\varphi}) &= 0 \text{ on } \partial\Omega \\ \frac{\partial}{\partial n}(\varphi + \tilde{\varphi}) &= 0 \text{ on } \partial\Omega. \end{aligned}$$

It is easily seen that in the second and third cases $\varphi(\boldsymbol{x}, \boldsymbol{y})$ was chosen as a solution of the given problem with the right hand side as Dirac delta at $\boldsymbol{y}$ and its generalized derivative, i.e., dipole (but for $a = a_0 =$ constant). That is why we recover $u(\boldsymbol{y})$ and $\partial u/\partial x(\boldsymbol{y})$. The function $\tilde{\varphi}$ was picked such that $(\varphi + \tilde{\varphi})$ contains the appropriate singularity of φ but vanishes with its normal derivative on $\partial\Omega$ so that no boundary terms result from integrating by parts. The last term in our expression takes into account possible dependence of the coefficient $a(\boldsymbol{x})$ on $\boldsymbol{x}$ and because of its strong singularity (r^{-3}) it should be understood in the sense of the Cauchy Principal Value.

The predicted rate of convergence of extracted quantities is equal to the double rate of convergence of $\hat{u}$ in the energy norm. This means that, for instance, for bilinear elements we should get (h^2) error which is superior to that of directly computed derivatives.

The method gives very good accuracy and has several attractive features. However, it has some important drawbacks that essentially limit its applications:

- it requires calculation of strongly singular integrals over the entire domain,
- there are difficulties with constructing a function $\tilde{\varphi}$ for complicated domains; moreover, the more irregular the $\tilde{\varphi}$, the worse the quality of results given by the extractions procedure,
- there is a possibility to avoid the above two problems by restricting calculation to subdomains of a chosen regular shape (for instance squares); however, it turns out that such a procedure deteriorates the results very significantly;
- accuracy deteriorates also very rapidly as one approaches the boundary of Ω, and
- the approach is limited to elliptic operators for which we know fundamental solutions (at least after averaging coefficients).

Recovery of Boundary Fluxes

As we have already mentioned, it can be very awkward to obtain good recovered approximations to fluxes near the boundary. The following method [44] is tailored toward deriving approximations to fluxes at the boundary.

Consider the problem

$$\left.\begin{aligned} -\Delta u + u &= f \quad \text{in} \quad \Omega \\ u &= 0 \quad \text{on} \quad \partial\Omega \end{aligned}\right\} \tag{194}$$

Suppose $\hat{u} \in \widehat{X} \subset X = \{v \in H^1(\Omega) \colon v = 0 \text{ on } \partial\Omega\}$, is the finite element approximation to u. Thus

$$\hat{u} \in \widehat{X} \colon \int_\Omega (\nabla\hat{u} \cdot \nabla v + \hat{u}v)\, d\boldsymbol{x} = \int_\Omega fv\, d\boldsymbol{x} \qquad \forall\, \boldsymbol{v} \in \widehat{X} \tag{195}$$

Let $\widehat{X}^0$ denote the set of shape functions which vanish at all the *interior nodes* on the mesh. Then for any $\varphi_j \in \widehat{X}^0$ we obtain from (194):

$$\int_\Omega (\nabla u \cdot \nabla \varphi_j + u\varphi_j)\, d\boldsymbol{x} = \int_\Omega f\varphi_j d\boldsymbol{x} + \oint_{\partial\Omega} \varphi_j \frac{\partial u}{\partial n}\, ds \tag{196}$$

or rearranging we obtain

$$\oint_{\partial\Omega} \varphi_j \frac{\partial u}{\partial n}\, ds = \int_\Omega (\nabla u \cdot \nabla \varphi_j + u\varphi_j)\, d\boldsymbol{x} - \int_\Omega f\varphi_j\, d\boldsymbol{x} \tag{197}$$

Replacing u by $\hat{u}$ on the right hand side gives the following statement for our approximation $g \approx \frac{\partial u}{\partial n}$ on $\partial\Omega$:

$$\oint_{\partial\Omega} \varphi_j g\, ds = \int_\Omega (\nabla \hat{u} \cdot \nabla \varphi_j + \hat{u}\varphi_j)\, d\boldsymbol{x} - \int_\Omega f\varphi_j\, d\boldsymbol{x} \qquad \forall\ \varphi_j \in \widehat{X}^0 \tag{198}$$

We now expand g in terms of the elements of $\widehat{X}^0$: $g(s) = \sum_{j=1}^{n} \alpha_k \varphi_k(s)$ and substitute into (198) to obtain the system:

$$M\boldsymbol{\alpha} = \boldsymbol{b} \tag{199}$$

where M is the $n \times n$ matrix with entries

$$M_{ij} = \oint_{\partial\Omega} \varphi_i(s)\varphi_j(s)\, ds \tag{200}$$

and $\boldsymbol{b}$ is the n vector with entries

$$b_i = \int_\Omega (\nabla \hat{u} \cdot \nabla \varphi_i + \hat{u}\varphi_i)\, d\boldsymbol{x} - \int_\Omega f\varphi_i\, d\boldsymbol{x} \tag{201}$$

In particular, b_i are rather easy to compute using the equations for the finite element approximation $\hat{u}$ *before* the boundary conditions are imposed. It can be shown [45] that the resulting scheme is superconvergent. The scheme is very inexpensive owing to the presence of the tridiagonal matrix M.

Summary and Conclusions

We have described several methods for postprocessing. The remarks made in the text show that there is a great danger that several of these methods will not work for general approximations. The most promising methods seem to be those based on superconvergence and projection methods in conjunction with the procedure described Cantin-Loubignac iteration. The use of the Cantin-Loubignac procedure is critical in the case of non-uniform meshes since otherwise the superconvergence techniques will fail. Meanwhile, using only projection techniques would result in very poor gradients near to the boundary.

The main drawback of these is the cost, but it is conceivable that by working *locally* one could significantly reduce the costs without sacrificing accuracy.

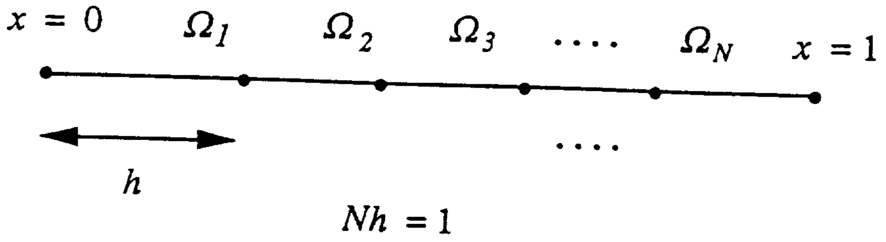


Figure 3.44: Uniform mesh in $\mathbb{R}^1$.

Extraction methods are rather unsuited for use with a general purpose package owing to their high dependence on the features of each particular problem. Moreover, they are expensive to implement and to use.

Boundary flux recovery methods represent a very attractive method for dealing with recovery of fluxes on the boundary and again is valid with general types of meshes. Thus the most favorable techniques can be classified according to the portion of the domain in which they are to be used:

1. For postprocessing on the interior of the mesh, use superconvergence and projection methods in conjunction with Cantin-Loubignac iteration and experiment to see if they can be applied locally.

2. For postprocessing on the boundary, use the boundary flux recovery technique.

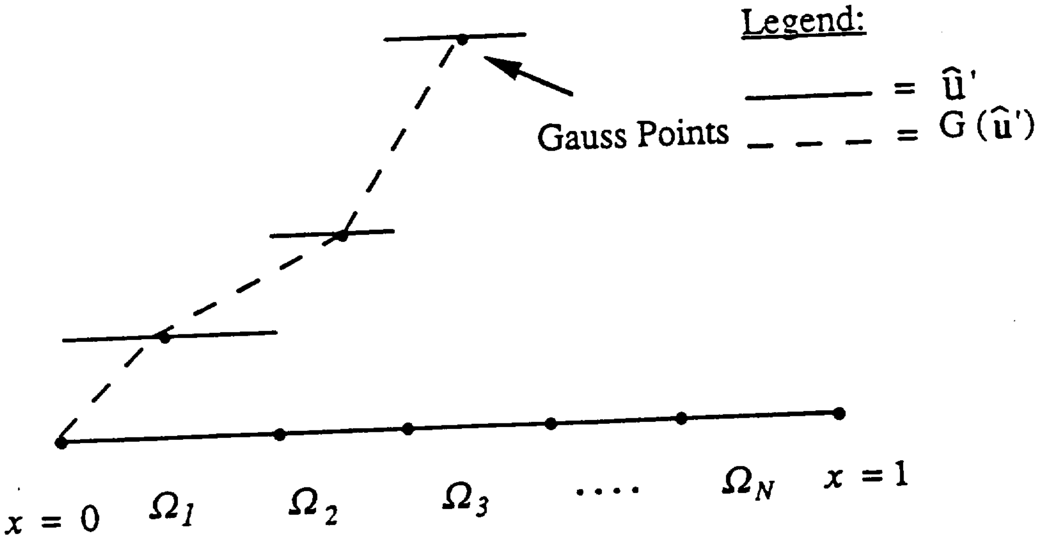


Figure 3.45: Postprocessing when $p = 1$.

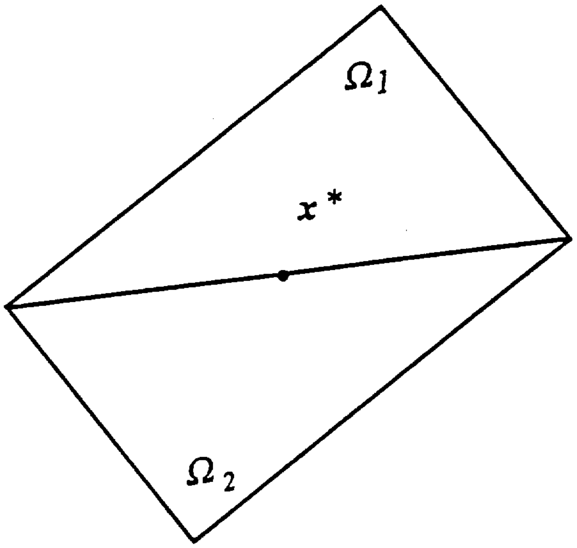


Figure 3.46: Pair of linear triangles.

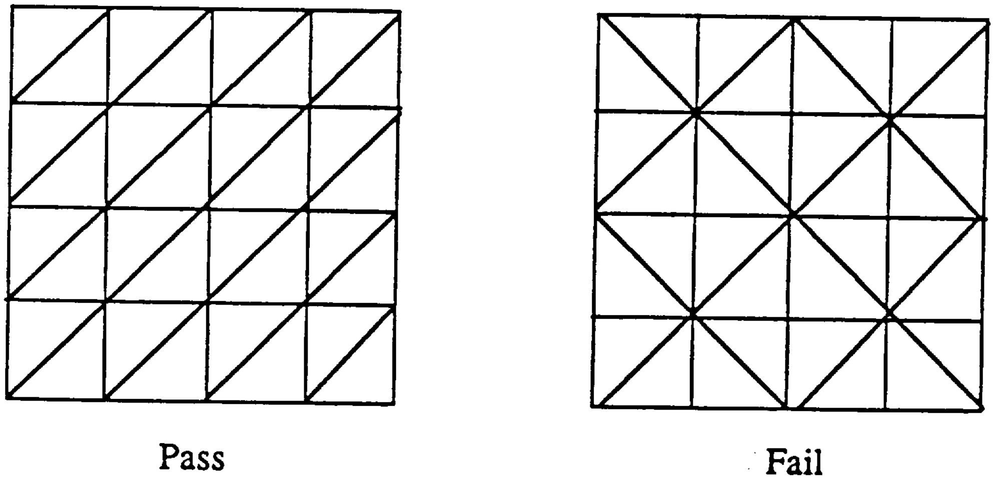


Figure 3.47: Meshes satisfying and failing the "six triangle" property.

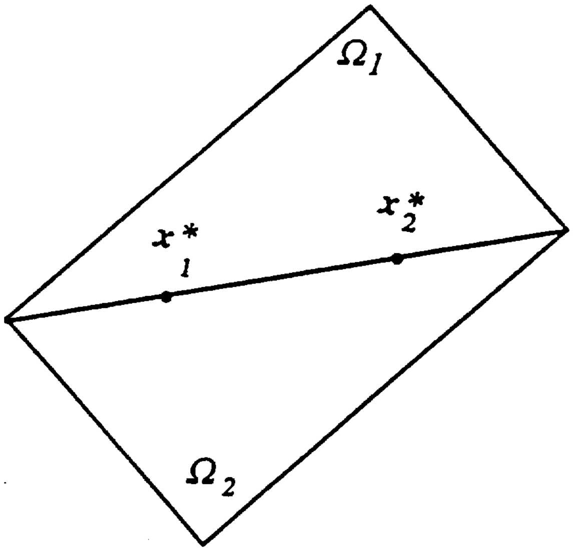


Figure 3.48: Stress points for quadratic elements.

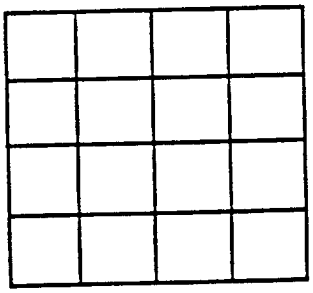


Figure 3.49: Regular mesh of quadrilaterals.

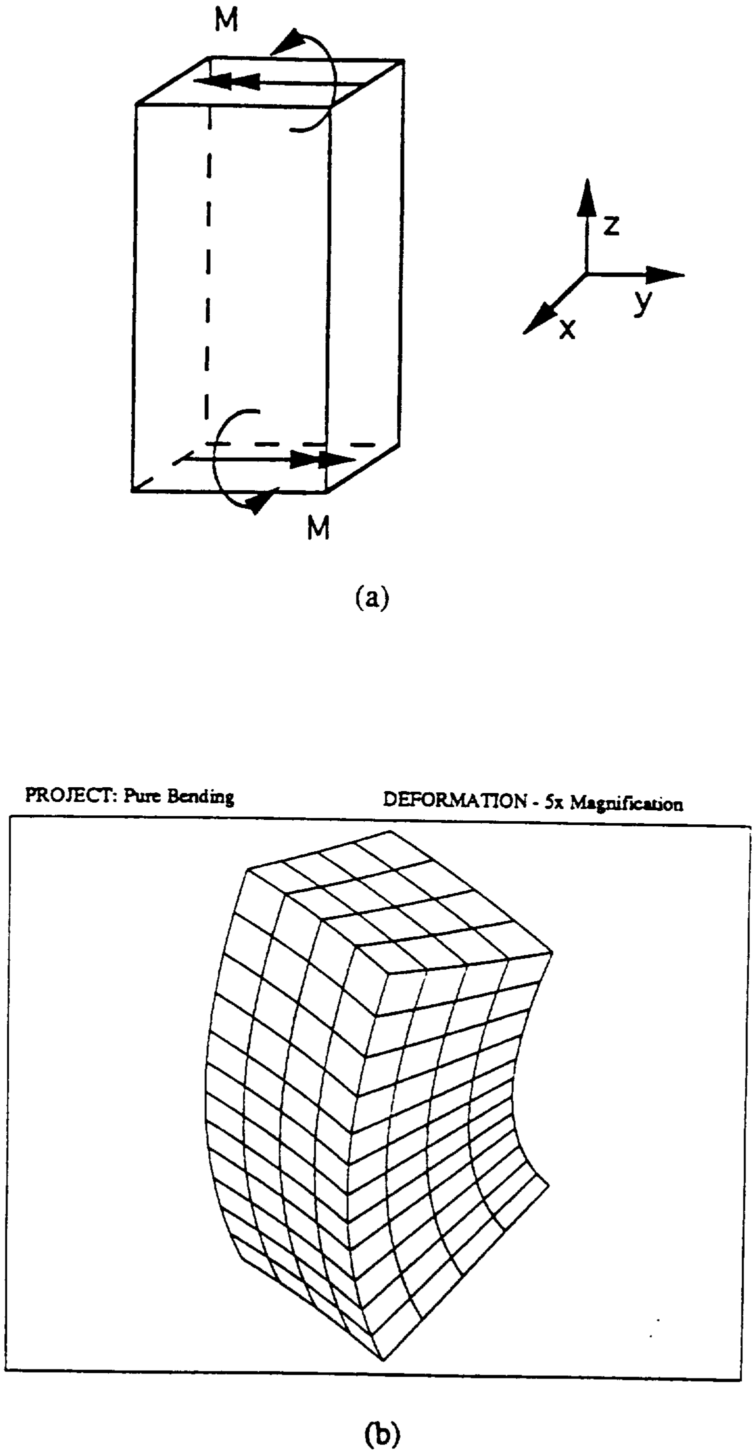
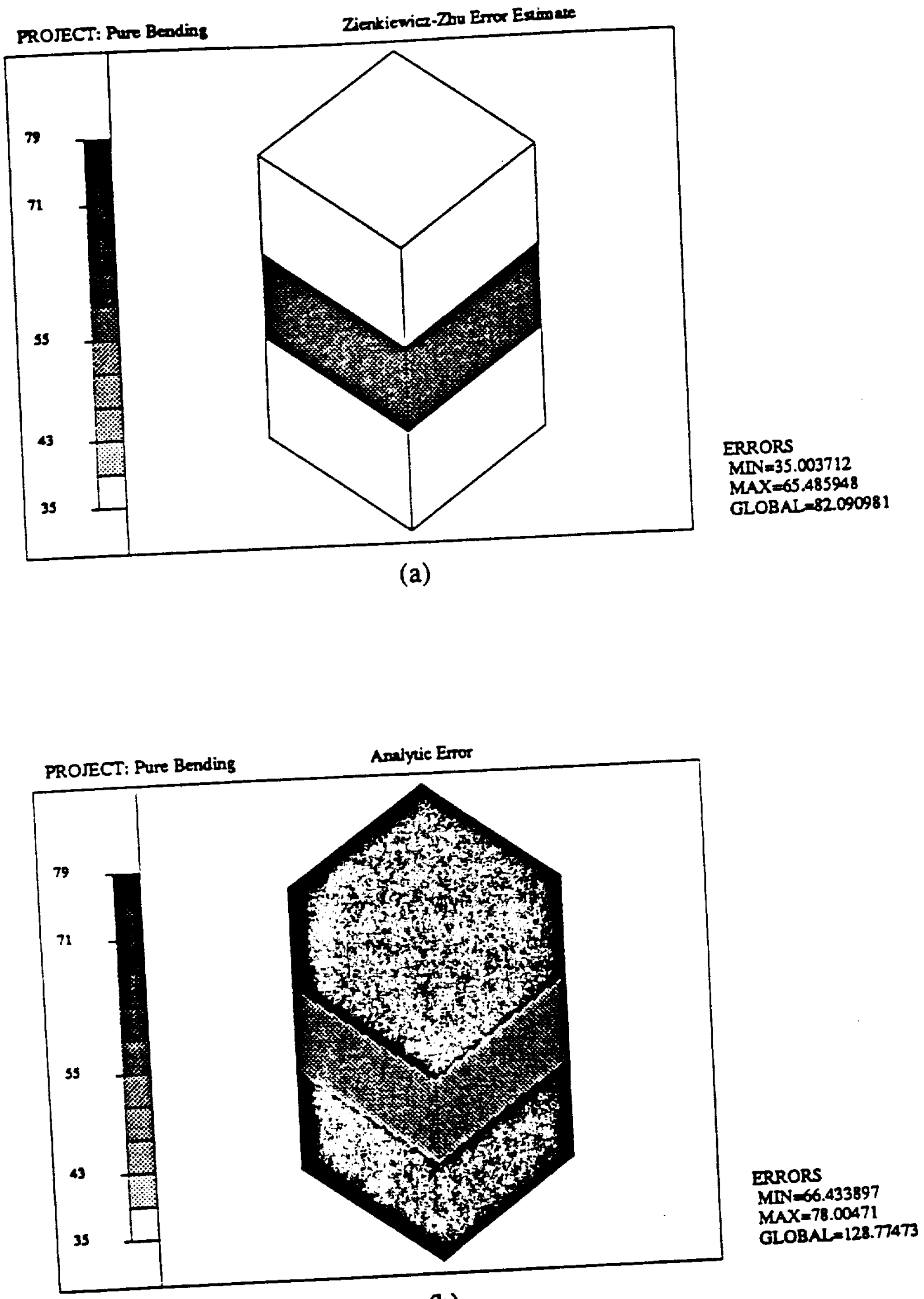


Figure 3.50: Pure bending of a prismatic beam. (a) Beam with applied moment, (b) deformed shape of beam (magnified five times).



(a)

(b)

Figure 3.51: Pure bending of a prismatic beam analyzed using a mesh of three linear elements. (a) Error estimates calculated using the Zienkiewicz-Zhu method, (b) analytic errors (measured in the energy norm).

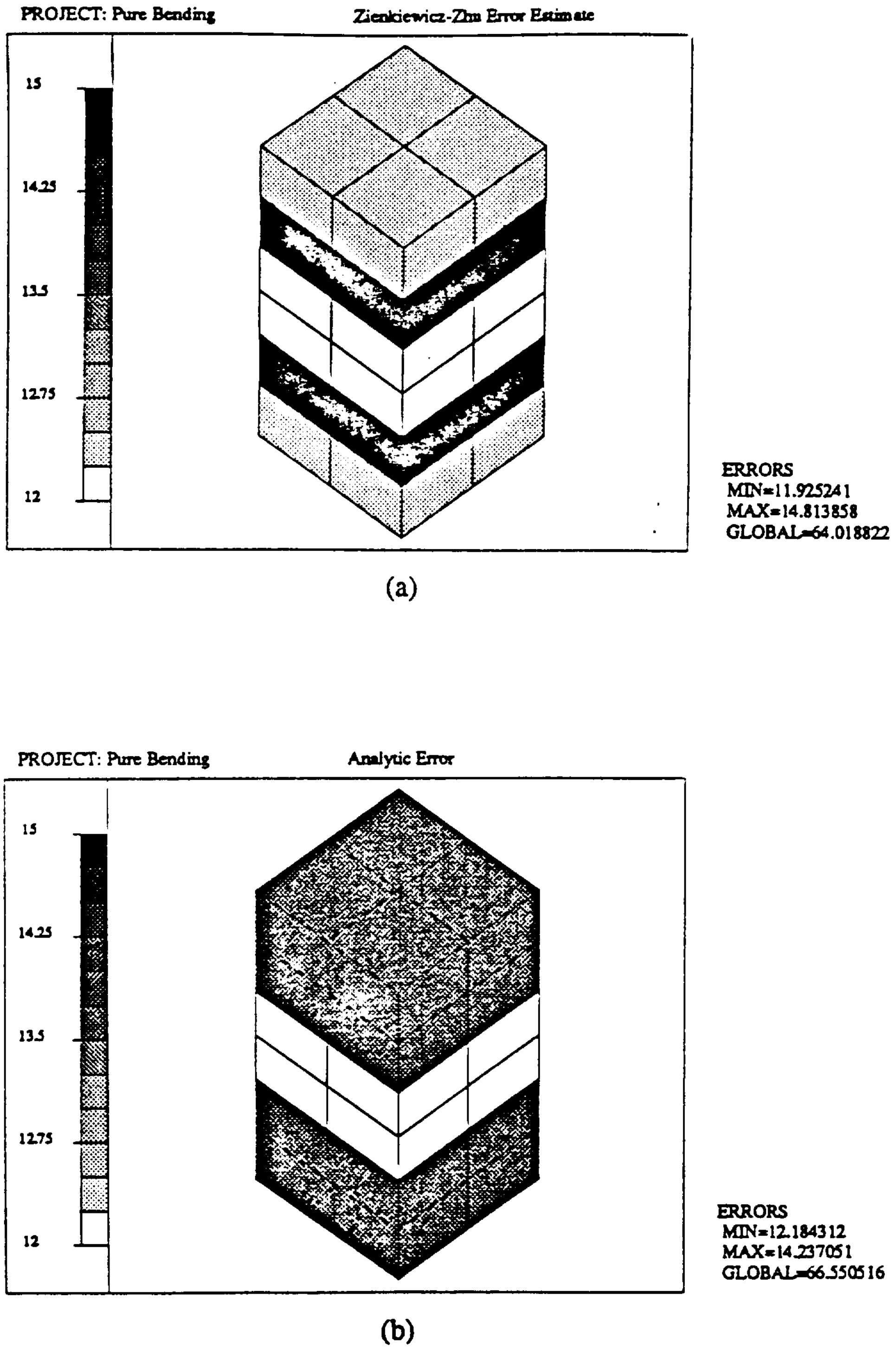


Figure 3.52: Pure bending of a prismatic beam analyzed using a mesh of 24 linear elements. (a) Error estimates calculated using the Zienkiewicz-Zhu method, (b) analytic errors (measured in the energy norm).

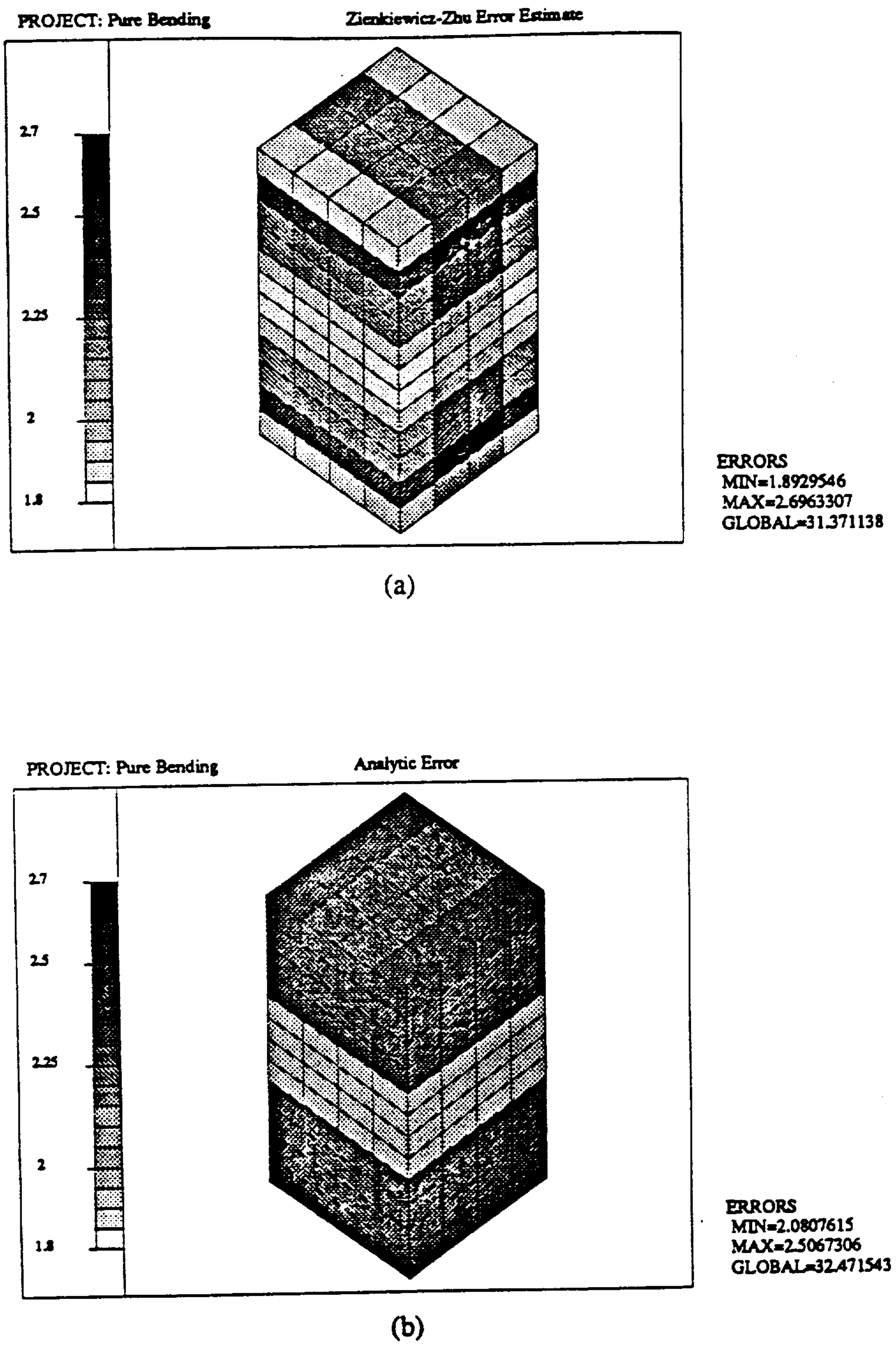
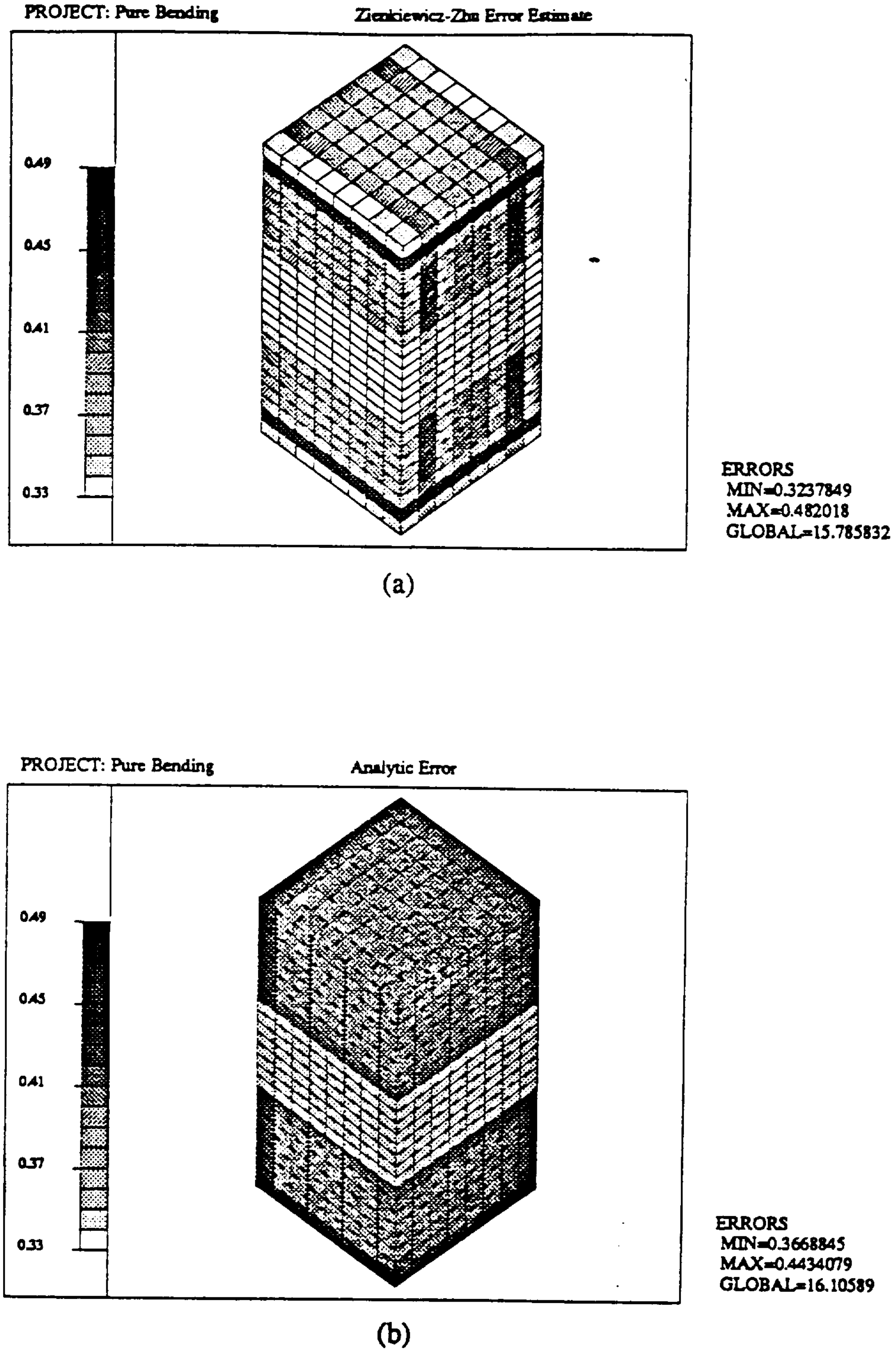


Figure 3.53: Pure bending of a prismatic beam analyzed using a mesh of 192 linear elements. (a) Error estimates calculated using the Zienkiewicz-Zhu method, (b) analytic errors (measured in the energy norm).



(a)

(b)

Figure 3.54: Pure bending of a prismatic beam analyzed using a mesh of 1536 linear elements. (a) Error estimates calculated using the Zienkiewicz-Zhu method, (b) analytic errors (measured in the energy norm).

3.3.2 Error Estimate Application: The Zienkiewicz-Zhu Technique

While the methods in the previous section were examined in the context of solution enhancement, they can also be applied to error estimation. Namely, an enhanced solution can be compared to the original solution to provide an estimate of the error in the original solution. For example, the Zienkiewicz-Zhu family of error estimates is based on superconvergence and projection methods. In this section, we present the results of a study of an enhanced version of the simple error estimator proposed by Zienkiewicz and Zhu [16] and analyzed by Ainsworth, et al. [38]. This technique was incorporated into an existing *hp*-adaptive finite element analysis code for testing. Following an outline of the technique, a few examples are shown for several types smooth problems.

For a governing differential equation and boundary conditions of the form

$$\begin{aligned} -\nabla \cdot (A\nabla u) + B \cdot \nabla u + Cu &= f \text{ in } \Omega \\ u &= f_0 \text{ on } \partial\Omega \end{aligned}$$

an estimate of the element-wise error is sought in the following semi-norm:

$$E(\Omega_k) = \left[\int_{\Omega_k} (\nabla u - \nabla u_h) \cdot [A(\nabla u - \nabla u_h)] \; d\Omega\right]^{\frac{1}{2}}$$

The error estimates are calculated by first calculating the finite element solution, $u_h \in V_h$, and then projecting the derivatives of u_h onto V_h to obtain G_h. Namely, $G_h \in V_h \times V_h$ satisfies

$$\min_{F_h \in V_h \times V_h} \int_\Omega [A(F_h - \nabla u_h)] \cdot (F_h - \nabla u_h)\, d\Omega$$

The Zienkiewicz-Zhu error estimate is then calculated according to:

$$E_h(\Omega) = \left[\sum_k \int_{\Omega_k} (G_h - \nabla u_h) \cdot [A(G_h - \nabla u_h)] \; d\Omega\right]^{\frac{1}{2}}$$

Sample Results

In order to compare the error estimates to the analytic errors, the effectivity indices are defined globally and elementwise as

$$\gamma_\Omega = \frac{E_h(\Omega)}{E(\Omega)} \quad \gamma_{\Omega_k} = \frac{E_h(\Omega_k)}{E(\Omega_k)}$$

where $E(\Omega) = \left[\sum_k [E(\Omega_k)]^2\right]^{\frac{1}{2}}$.

Example: Problem With Polynomial Solution

For this example, the loading function f and the boundary condition f_0 were chosen to correspond to the simple polynomial:

$$u = \left(x - \frac{1}{2}\right)^3 + \left(y - \frac{1}{2}\right)^3 - 1$$

Figure 3.55 shows a contour and a carpet plot of this function along with the finite element mesh used in this example. Figures 3.56, 3.57, and 3.58 show plots of the elementwise effectivity indices and the estimated error for the following forms of the differential equation:

$$\text{Figure 3.56:} \quad -\nabla \cdot \nabla u = f$$

$$\text{Figure 3.57:} \quad -\nabla \cdot \nabla u + u = f$$

$$\text{Figure 3.58:} \quad -\nabla \cdot \nabla u + \frac{\partial u}{\partial x} = f$$

As shown in these three figures, the estimated errors are quite accurate (the effectivity indices are close to one) for all three forms of the governing differential equation.

Example: Problem With Sinusoidal Solution

For this example, the loading function, f, and the boundary condition, f_0, were chosen to correspond to the analytic solution:

$$u = \sin(x + 3y) + e^x + y^2$$

Figure 3.59 shows a contour and a carpet plot of this function along with the finite element mesh used in this example. Figure 3.60 shows plots of the elementwise effectivity indices and the estimated error for the differential equation:

$$-\nabla \cdot \nabla u = f$$

Except for a few elements, the estimated errors in this example are also quite good.

Example: Problem With Arctangent Solution

For this example, the loading function, f, and the boundary condition, f_0, were chosen to correspond to the analytic solution:

$$u = \tan^{-1}(15x)$$

Figure 3.61 shows a contour and a carpet plot of this function along with the finite element mesh used in this example. In Fig. 3.62, plots of the elementwise effectivity indices and the estimated error are shown for the differential equation:

$$-\nabla \cdot \nabla u = f$$

For this example, the estimated errors were not as accurate as the previous two examples, possibly due to the limitations of the technique.

Example: Elasticity problem

For this example, a short cantilever beam was analyzed using the sequence of meshes provided in Ainsworth, et al. [38]. Figure 3.63 shows the geometry and loading of the cantilever beam while Figs. 3.64 and 3.65 show the sequence of six h-adapted meshes. These meshes were used to illustrate another (apparent) feature of this type of error estimate. Namely, it is possible to estimate the effectivity index of a given error estimate. This is done by first computing the finite element solution and error estimate on the original mesh and then globally h-refining the mesh and recalculating only the error estimate on the newly-refined mesh. The effectivity index appears to obey the relationship:

$$\gamma \left(= \frac{E_h \text{ (initial mesh)}}{E} \right) \approx \left[\frac{E_h \text{ (refined mesh)}}{E_h \text{ (initial mesh)}} \right]^r$$

where $r \approx 3$.

Furthermore, this relationship can be used to produce an improved error estimate according to:

$$E_h(\text{improved}) = \frac{[E_h(\text{original mesh})]^{r+1}}{[E_h(\text{refined mesh})]^r}$$

Figure 3.66 shows the effects of these relationships for the six meshes of the cantilevered beam problem. In this figure, the estimated errors and effectivity indices are shown for various values for the exponent r. Namely, curves "estim" and "effecØ" correspond to the unmodified error estimates while "estnew1" and "effec1" correspond to a value $r = 1$, "estnew2" and "effec2" to a value $r = 2$ and "estnew3" and "effec3" to a value $r = 3$. As shown in this figure, this technique can greatly enhance the quality of the error estimate although the added expense of obtaining solution and error estimates on additional meshes makes the technique impractical.

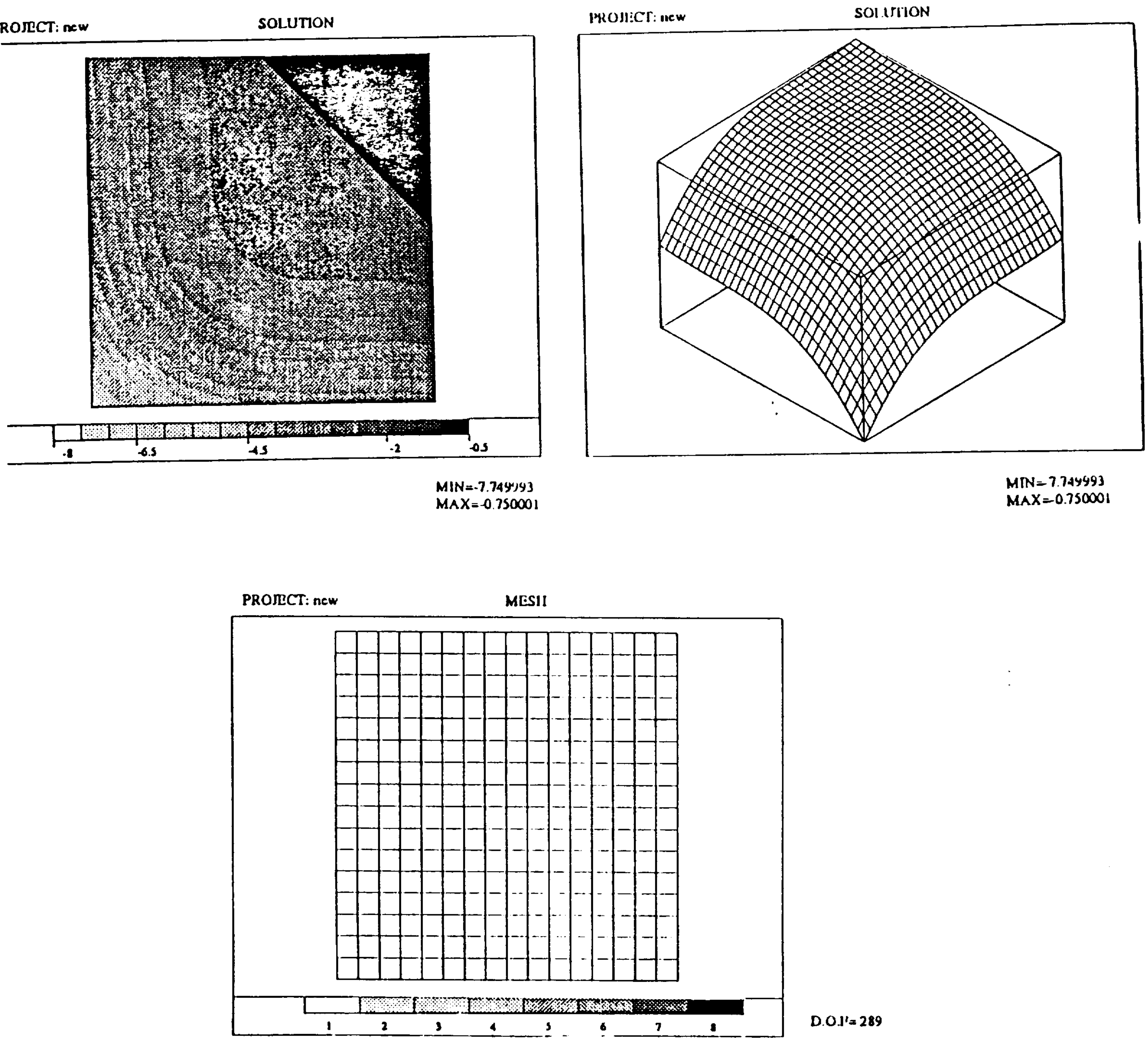


Figure 3.55: Plots of analytic solution, $u = (x - \frac{1}{2})^3 + (y - \frac{1}{2}) - 1$ and the finite element mesh used in the first example.

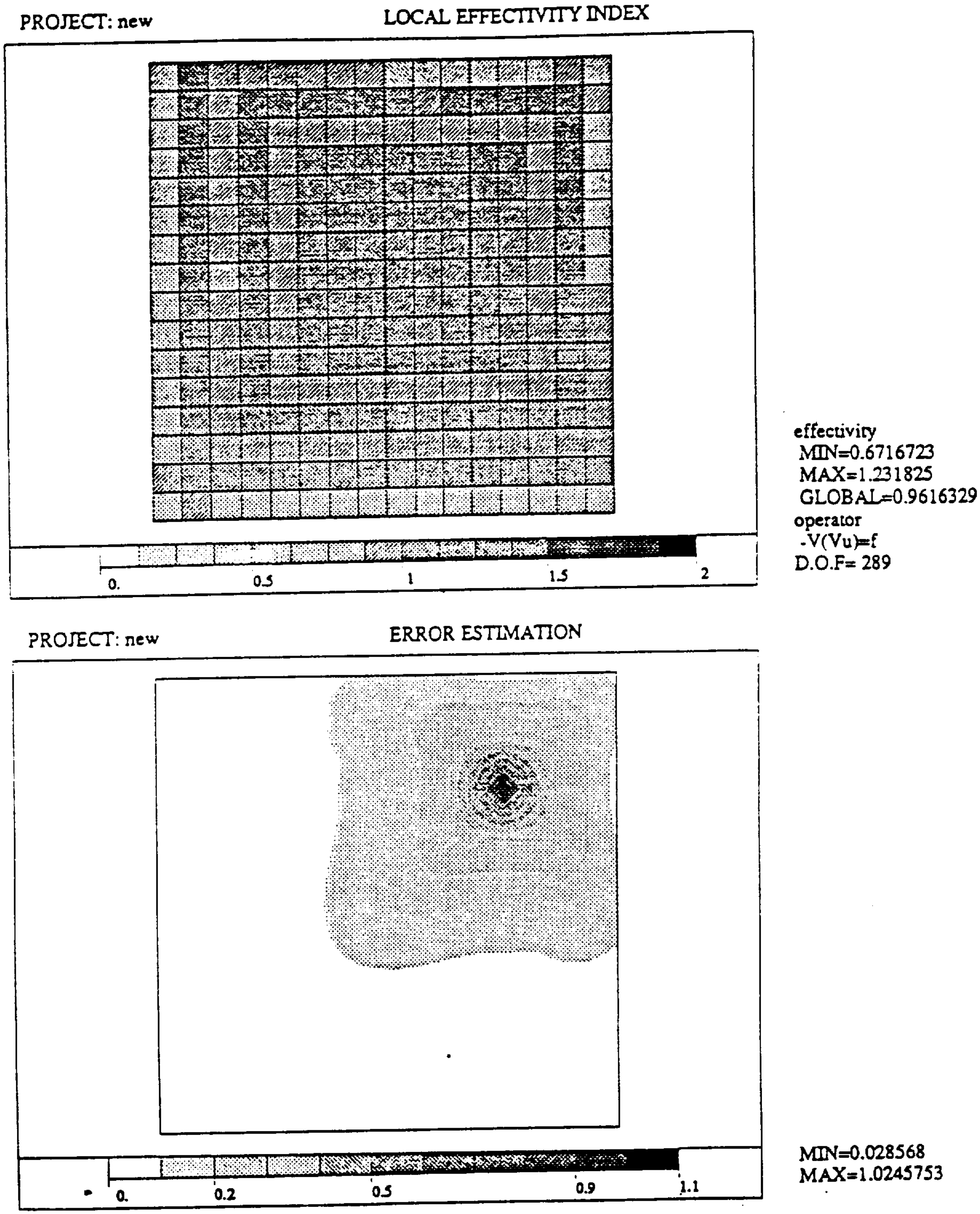


Figure 3.56: Local effectivity indices and estimated error for example with polynomial solution with the governing equation:

$$-\nabla \cdot \nabla u = f$$

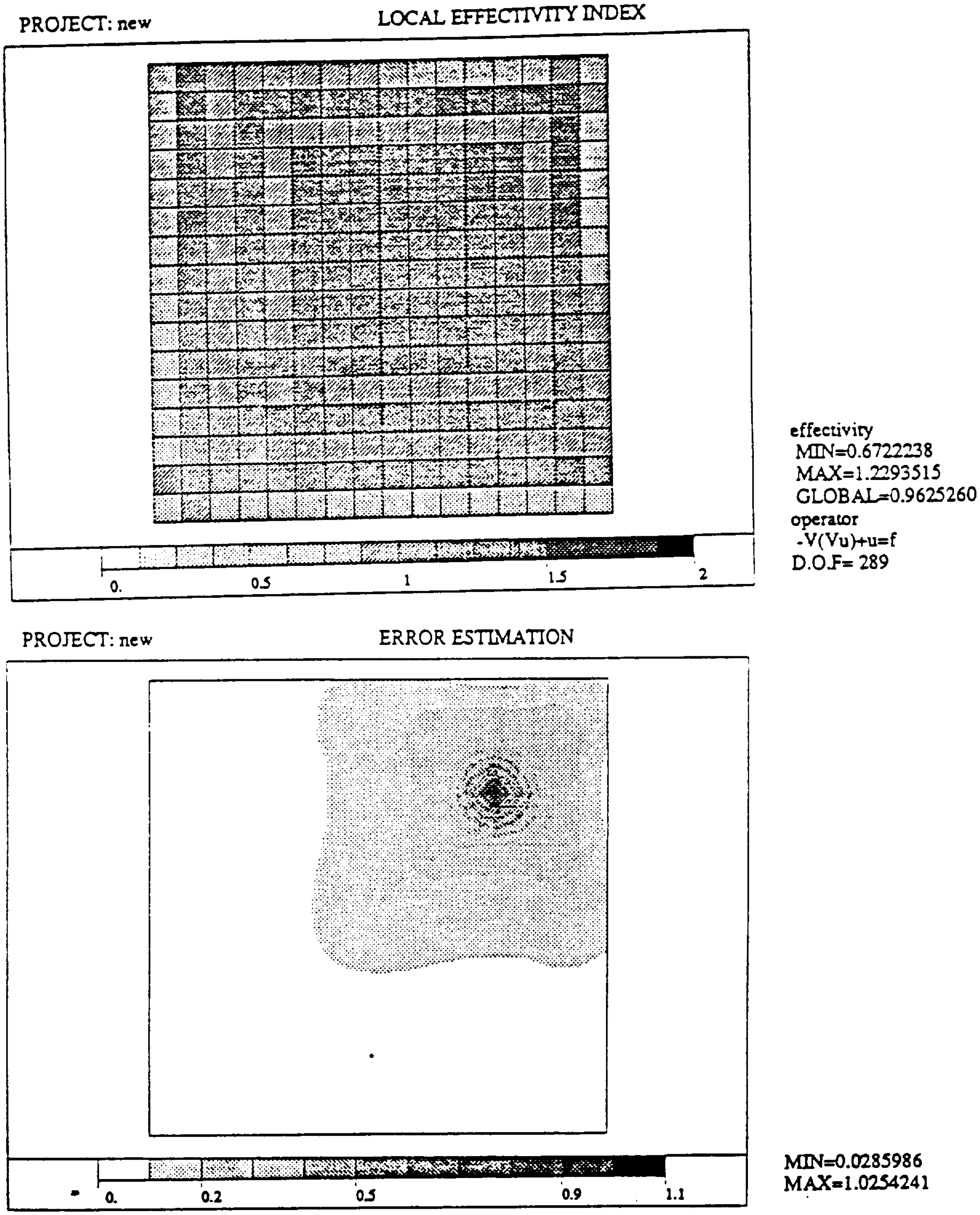


Figure 3.57: Local effectivity indices and estimated error for example with polynomial solution using the governing equation:

$$-\nabla \cdot \nabla u + u = f$$

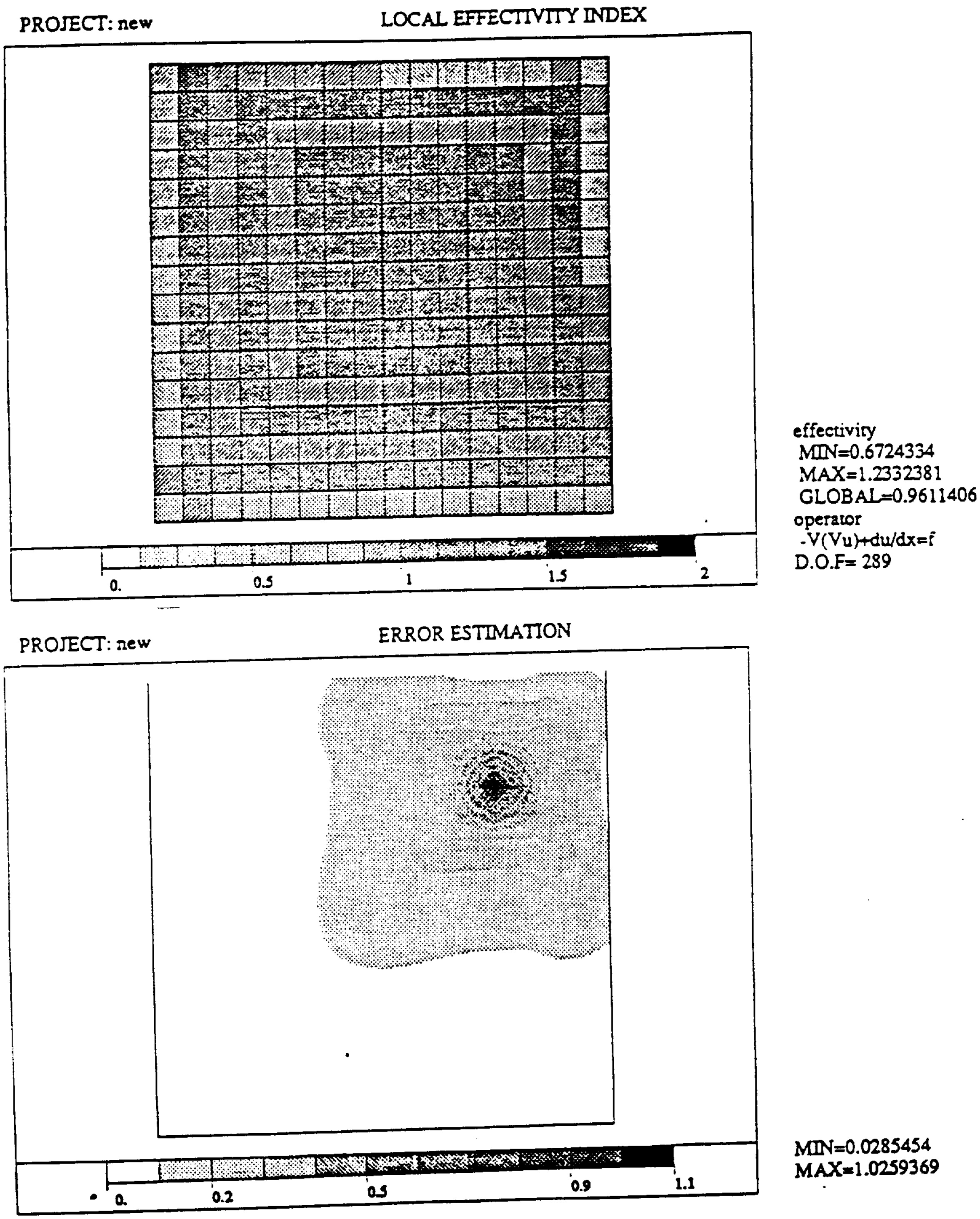


Figure 3.58: Local effectivity indices and estimated error for example with polynomial solution using the governing equation:

$$-\nabla \cdot \nabla u + \frac{\partial u}{\partial x} = f$$

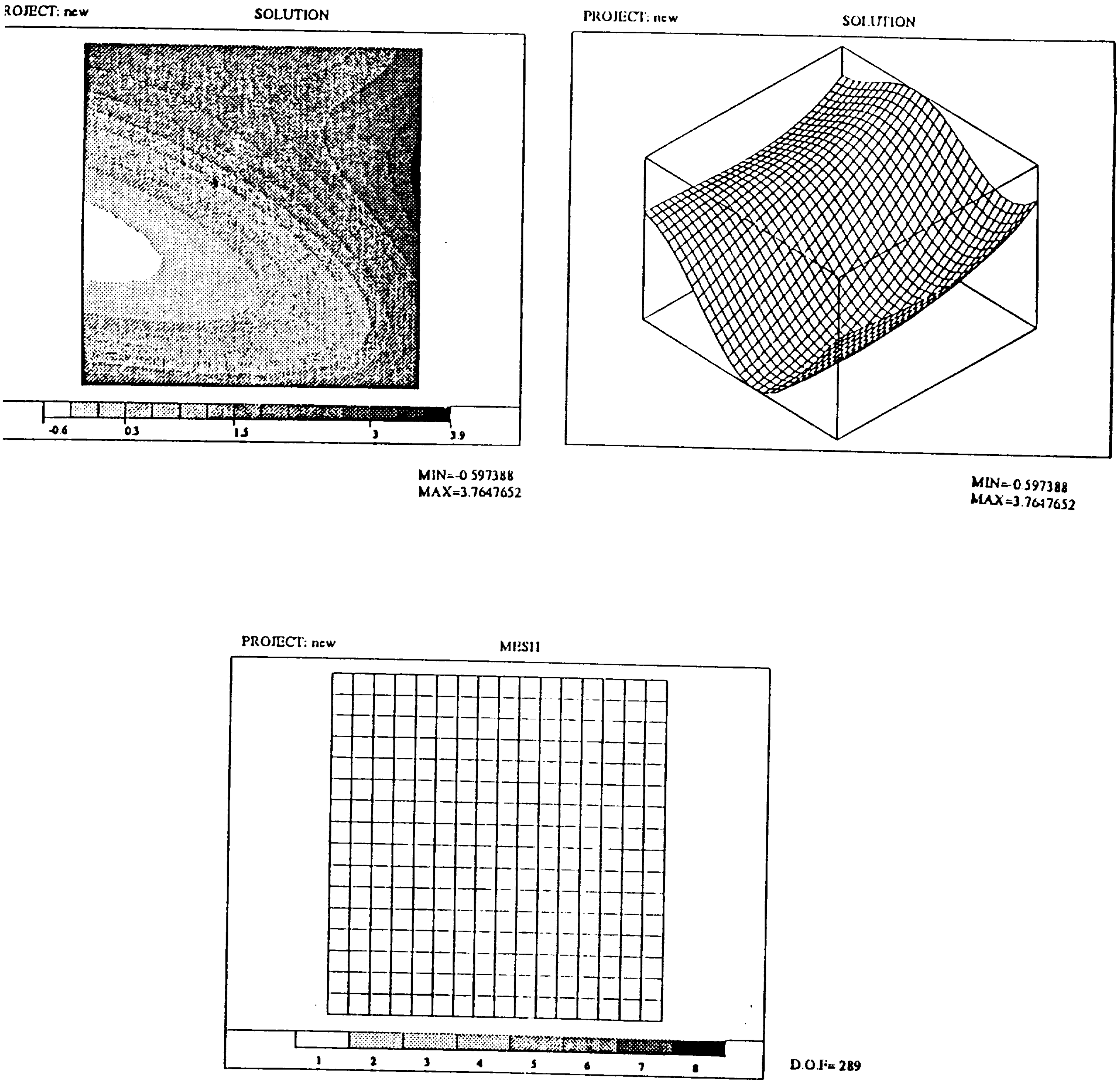


Figure 3.59: Plots of the analytic solution, $u = \sin(x + 3y) + e^x + y^2$ and the finite element mesh used in the second example.

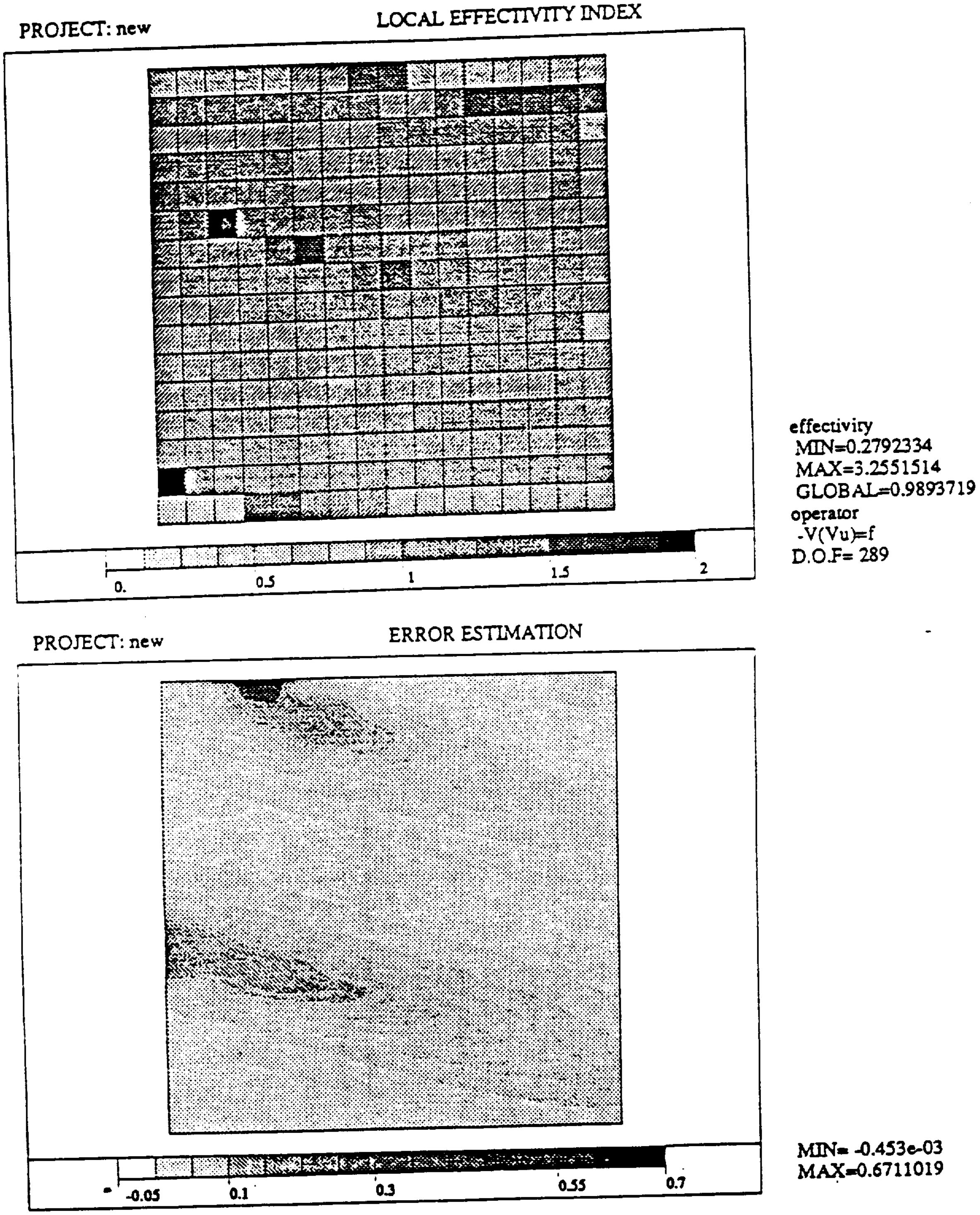


Figure 3.60: Local effectivity indices and estimated error for the second example with the governing equation:

$$-\nabla \cdot \nabla u = f$$

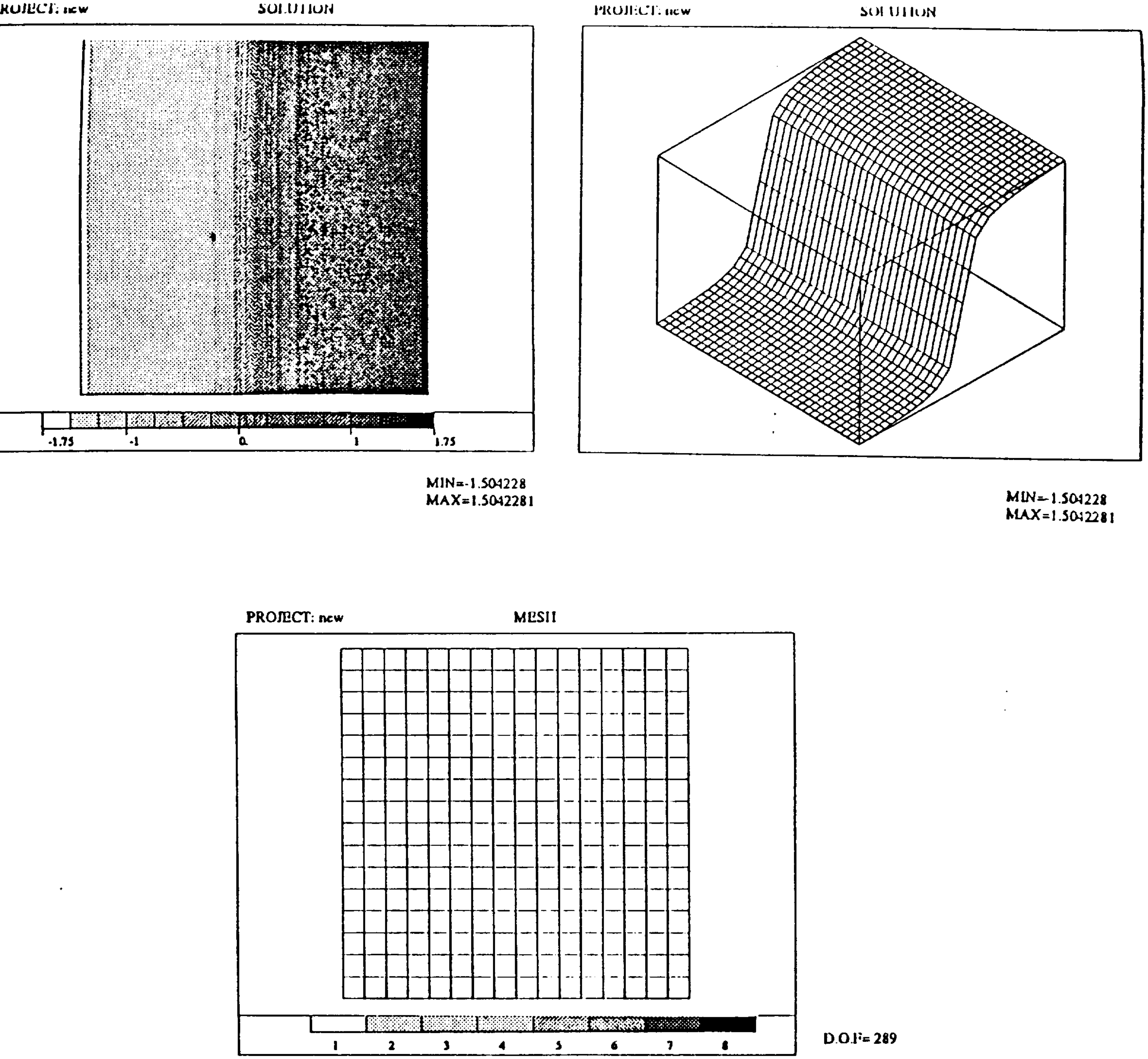


Figure 3.61: Plots of the analytic solution, $u = \tan^{-1}(15x)$ and the finite element mesh used in the third example.

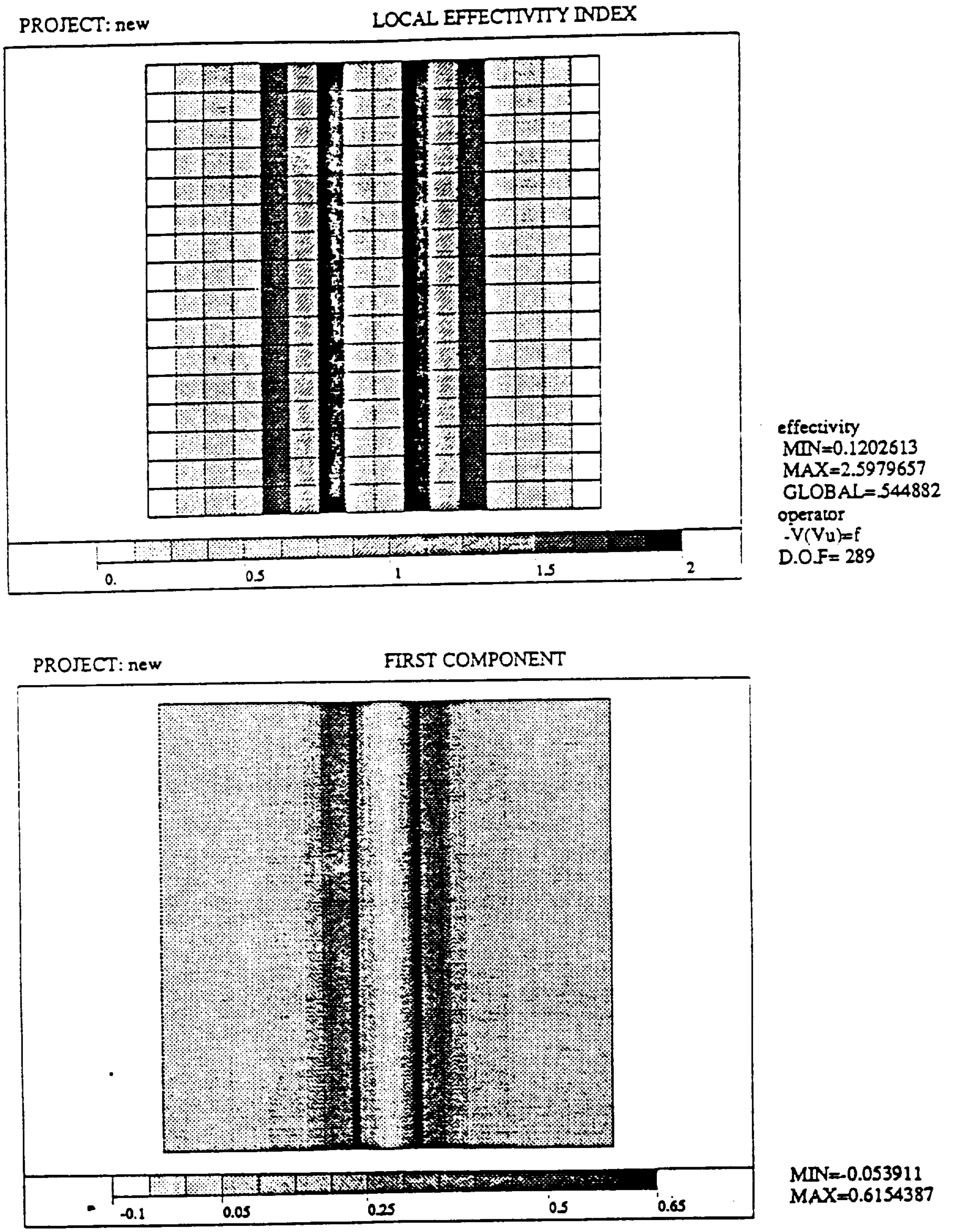


Figure 3.62: Local effectivity indices and estimated error for the third example with the governing equation:

$$-\nabla \cdot \nabla u = f$$

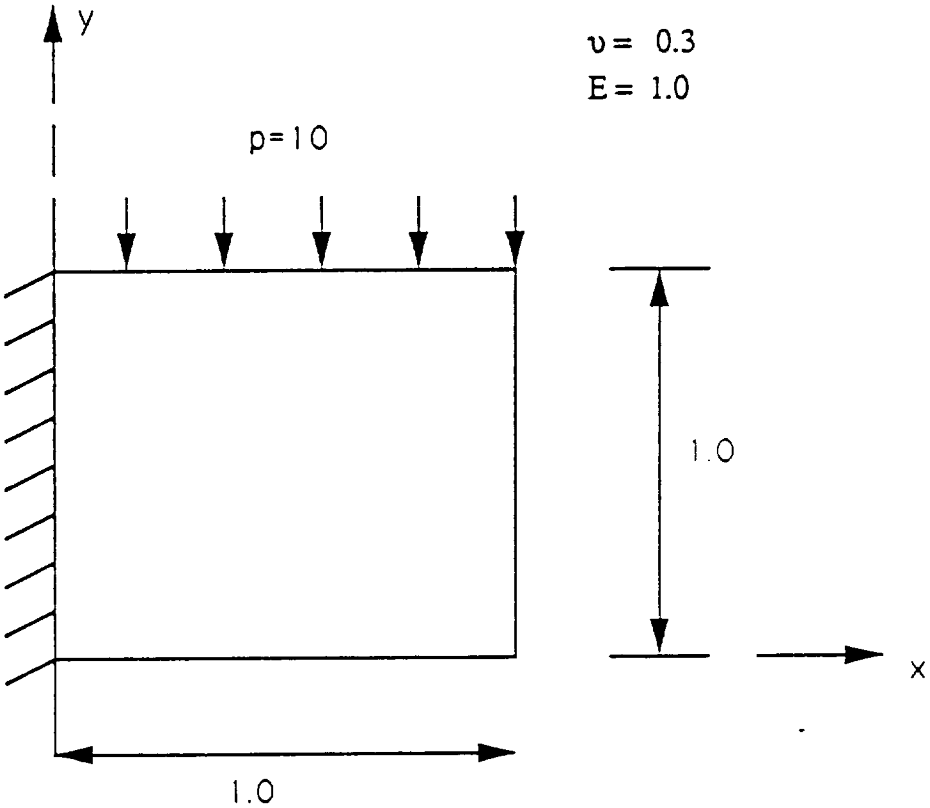


Figure 3.63: Geometry and loading of short cantilever beam used in the fourth example.

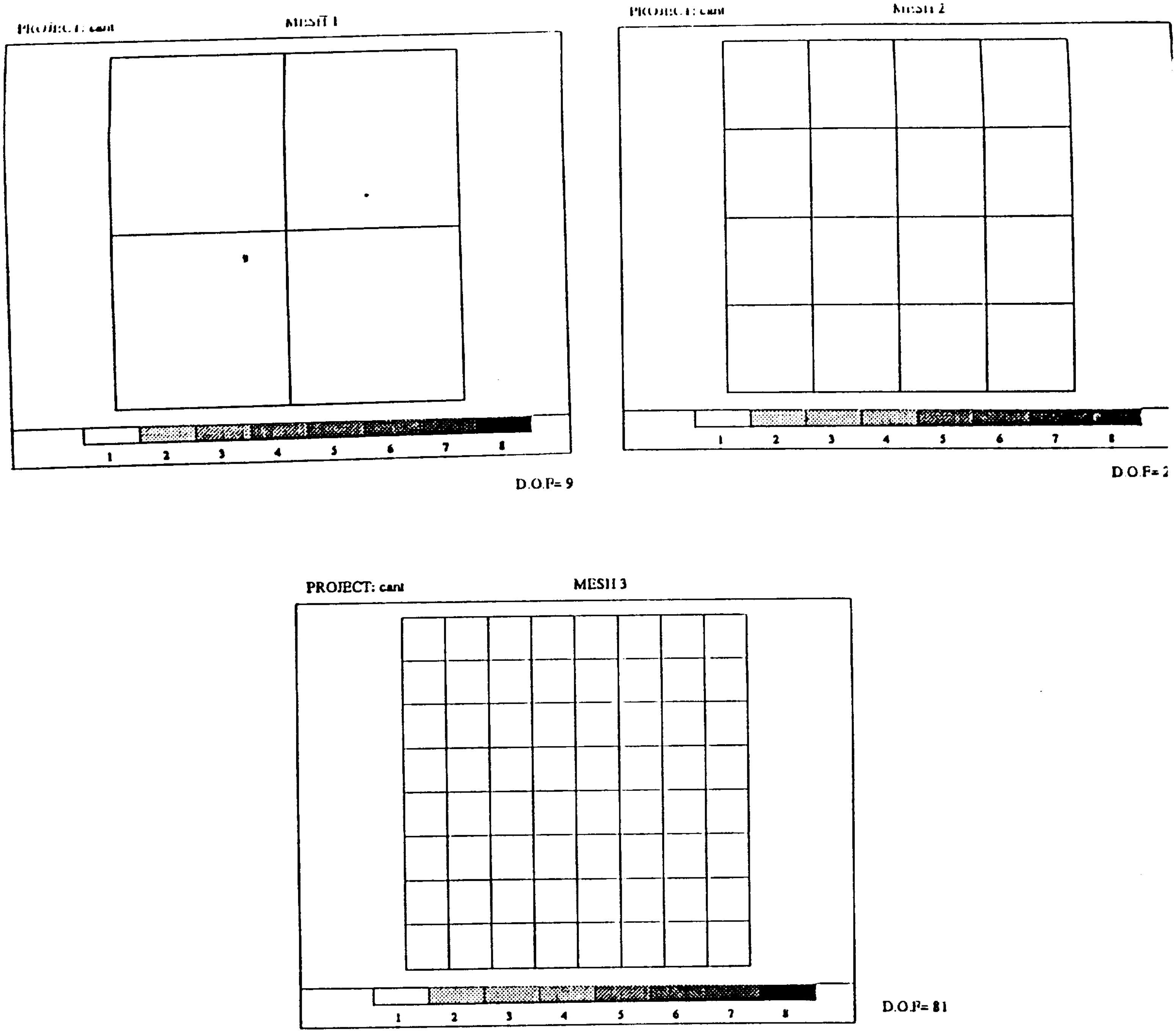


Figure 3.64: Meshes 1, 2, and 3 for the cantilevered beam problem.

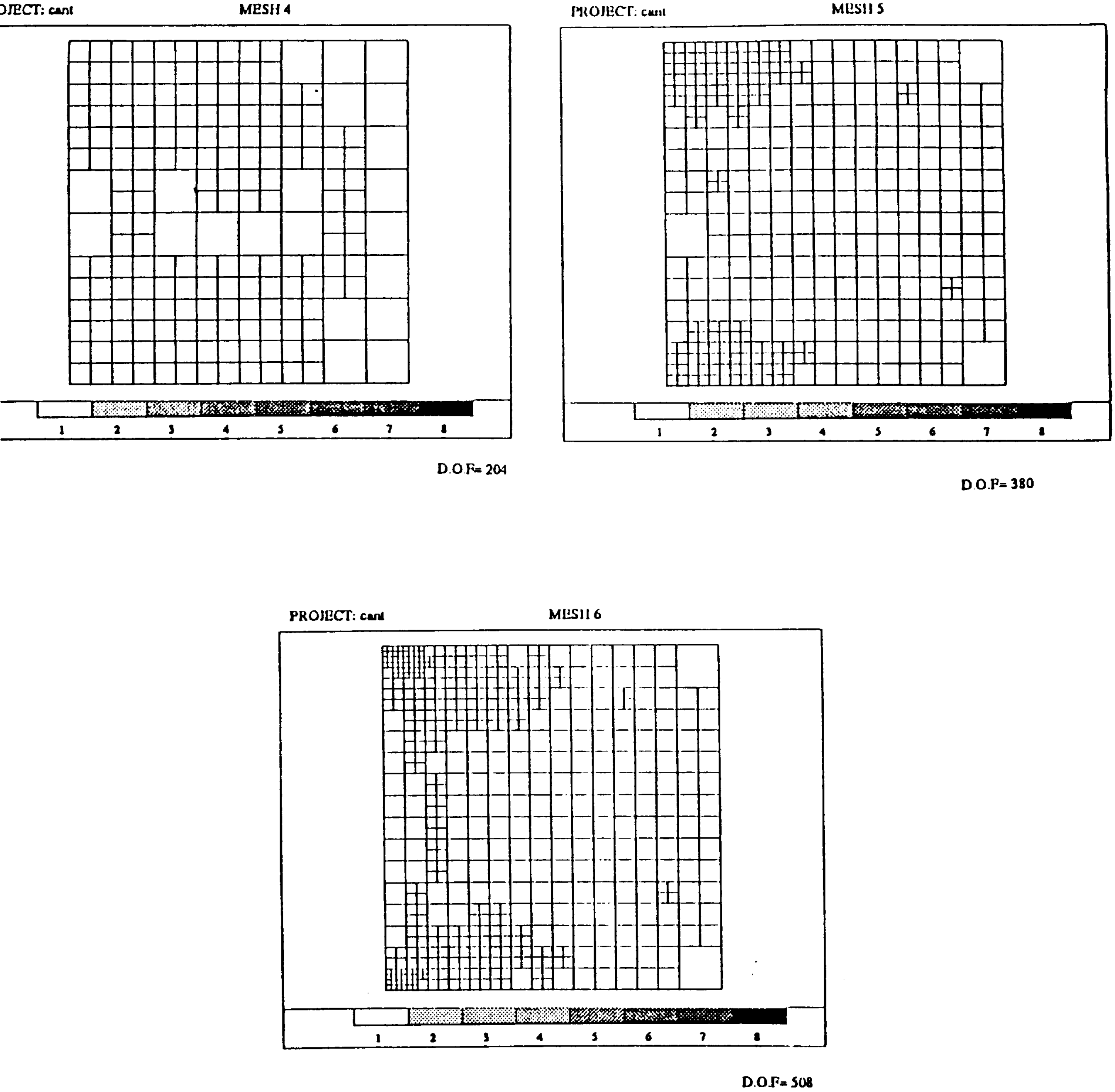
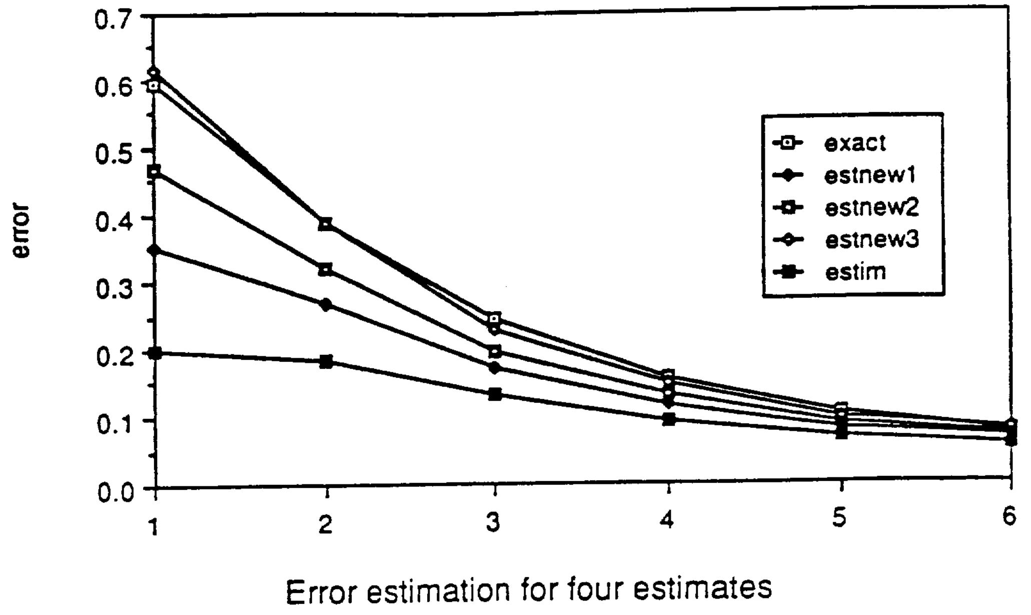


Figure 3.65: Meshes 4, 5, and 6 for the cantilever beam problem.



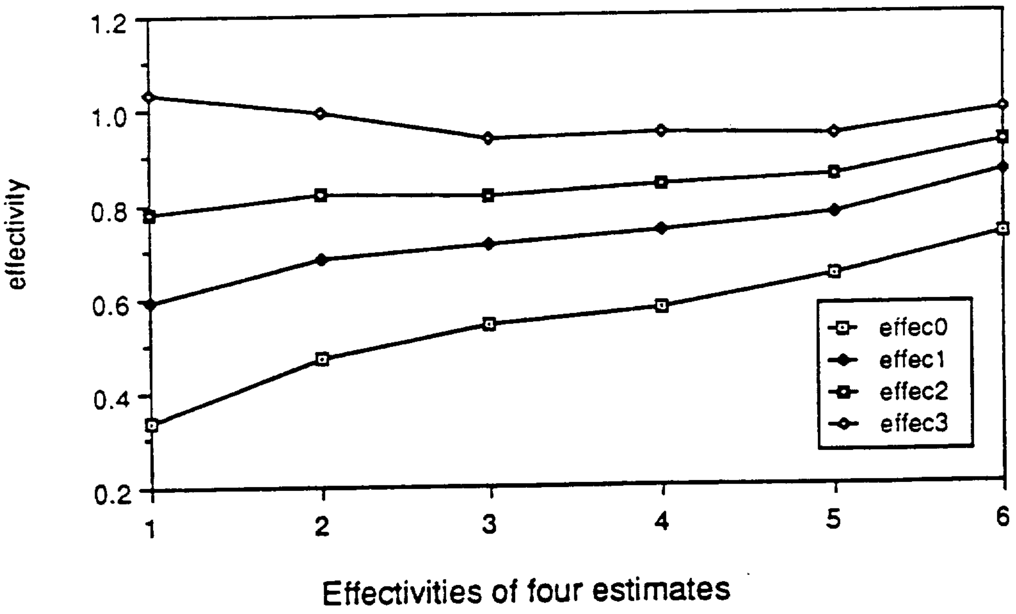


Figure 3.66: Error estimates and effectivity indices for the 6 meshes for the cantilevered beam example. Shown are results using different exponents for r for improving the quality of the error estimates.

3.3.3 Error Estimation and Solution Enhancement: The Projection Method with Cantin-Loubignac Iteration

In the previous section, error estimation results were presented using the technique proposed by Zienkiewicz and Zhu [16] and analyzed by Ainsworth, et al. [38]. This technique was used to apply projection methods to Poisson problems and problems in elasticity in order to obtain improved stresses as well as estimates of the error in the approximate solution. In addition, Section 3.3.1 introduced a technique called Cantin-Loubignac iteration which can be used to improve the basic error estimate. This technique is discussed further in this section.

First we recall that we seek a solution to the differential equation

$$-\nabla \cdot (A \nabla u) + B \cdot \nabla u + Cu = f \tag{202}$$

in the domain Ω, subject to the boundary conditions

$$\begin{aligned} u &= u_0 \quad \text{on} \quad \Gamma_D \\ A\frac{\partial u}{\partial n} &= t \quad \text{on} \quad \Gamma_t \end{aligned}$$

such that $\overline{\Gamma}_D \cup \overline{\Gamma}_t = \partial\Omega$ and $\boldsymbol{n}$ is the outward-directed boundary normal.

The variational form of this problem may be stated: Find $u \in V_0(\Omega) + \{u_0\}$ such that

$$B(u,v) = F(v) \qquad \forall\, v \in V_0(\Omega) \tag{203}$$

where $V_0(\Omega) = \{v\colon v \in H^1(\Omega),\, \gamma v = 0 \text{ on } \Gamma_D\}$ and

$$\begin{aligned} B(u,v) &= \int_\Omega [A\nabla u \cdot \nabla v + B \cdot \nabla u v + Cuv] d\Omega \\ F(v) &= \int_\Omega f v d\Omega + \int_{\Gamma_t} t v dS \end{aligned} \tag{204}$$

The basic Zienkiewicz-Zhu error estimate is obtained by first solving the approximate problem for the finite element solution, u_h

$$B\,(u_h, v_h) = F\,(v_h) \qquad \forall\, v_h \in V_h \subset V_0 \tag{205}$$

The projection problem is then solved to obtain the enhanced solution derivatives, $\boldsymbol{G}_h$

$$\int_\Omega \boldsymbol{G}_h \cdot \boldsymbol{v}_h d\Omega = \int_\Omega \nabla u_h \cdot \boldsymbol{v}_h d\Omega \qquad \forall\, \boldsymbol{v}_h \in V_h \times V_h \tag{206}$$

and the Zienkiewicz-Zhu estimate is obtained

$$E_{zz}(\Omega) = \left[\sum_k \int_{\Omega_k} (\boldsymbol{G}_h - \nabla u_h) \cdot A (\boldsymbol{G}_h - \nabla u_h) \, d\Omega \right]^{\frac{1}{2}} \tag{207}$$

This estimate can be improved by using the enhanced derivatives to construct a correction problem

$$B(\delta, v_h) = F(v_h) - \int_\Omega (A \boldsymbol{G}_h \cdot \nabla v_h) \, d\Omega \tag{208}$$

for $\delta \in V_h$ and $\forall \ v_h \in V_h$.

The approximate solution is then enhanced by adding the correction

$$u_h^+ = u_h + \delta \tag{209}$$

If the size of the correction, $||\delta||$, is not small, the enhanced solution derivatives, $\boldsymbol{G}_h$, are recalculated using equation (206) with the updated solution, u_h^+, and improved estimate using equation (207), and then a new correction using equation (208). This cycle can be repeated until $||\delta||$ is sufficiently small.

As an example of this technique, we analyzed the plane elasticity problem of a short cantilevered beam from the previous section using the sequence of meshes provided in Ainsworth, et al. [38]. Figure 3.63 shows the geometry and loading of the beam while Figs. 3.64 and 3.65 show the sequence of six h-adapted meshes. The above iterative procedure was performed until $||\delta||$ was six orders of magnitude smaller than $||u^+||$. Typically, only six or seven iterations were necessary. Figure 3.67a compares the modified error estimates to the standard Zienkiewicz estimates and actual errors (taken from [38]) for the six meshes. Similarly, Fig. 3.67b compares the effectivity indices (error estimate/actual error) for the two estimates.

As these figures show, the modified technique provides an improved estimate in all of the cases. Unfortunately, each iteration of this technique requires the solution of a correction problem that is just as expensive as the original solution. Thus, with the added cost of the projection problems, this technique is impractical on a global (meshwise) scale. In the next section, we examine the consequences of using this technique on a patchwise basis.

3.3.4 Localization Study of Solution Enhancement Techniques

Introduction

The purpose of this work was to examine the various types of postprocessing schemes suggested by a survey described in Section 3.3.1. In particular, the suggested techniques were studied in the context of finite element approximations and applied to one element at a time instead of to the entire mesh.

In order to simplify our explanation, we employ a very simple problem to illustrate the techniques.

Let $\Omega \subset \mathbb{R}^2$ be an open bound domain with Lipschitzian boundary $\partial\Omega$. For given data $f \in L^2(\Omega)$, and $g \in L^2(\Gamma_N)$ we seek the solution u of

$$\begin{aligned} -\nabla^2 u &= f \text{ in } \Omega \\ u &= 0 \text{ on } \Gamma_D \\ \frac{\partial u}{\partial n} &= g \text{ on } \Gamma_N \end{aligned} \qquad (210)$$

where $\Gamma_D \cap \Gamma_N$ is empty, Γ_D is non-empty and $\partial\Omega = \overline{\Gamma_D \cup \Gamma_N}$.

The variational form of (210) is to find

$$u \in X : B(u,v) = L(v) \quad \forall v \in X \qquad (211)$$

where

$$\left.\begin{aligned} X &= \{v \in H^1(\Omega) : \gamma v = 0 \text{ on } \Gamma_D\} \\ B(u,v) &= \int_\Omega \nabla u \cdot \nabla v d\boldsymbol{x} \\ L(v) &= \int_\Omega f v d\boldsymbol{x} + \int_{\Gamma_N} g v ds \end{aligned}\right\} \qquad (212)$$

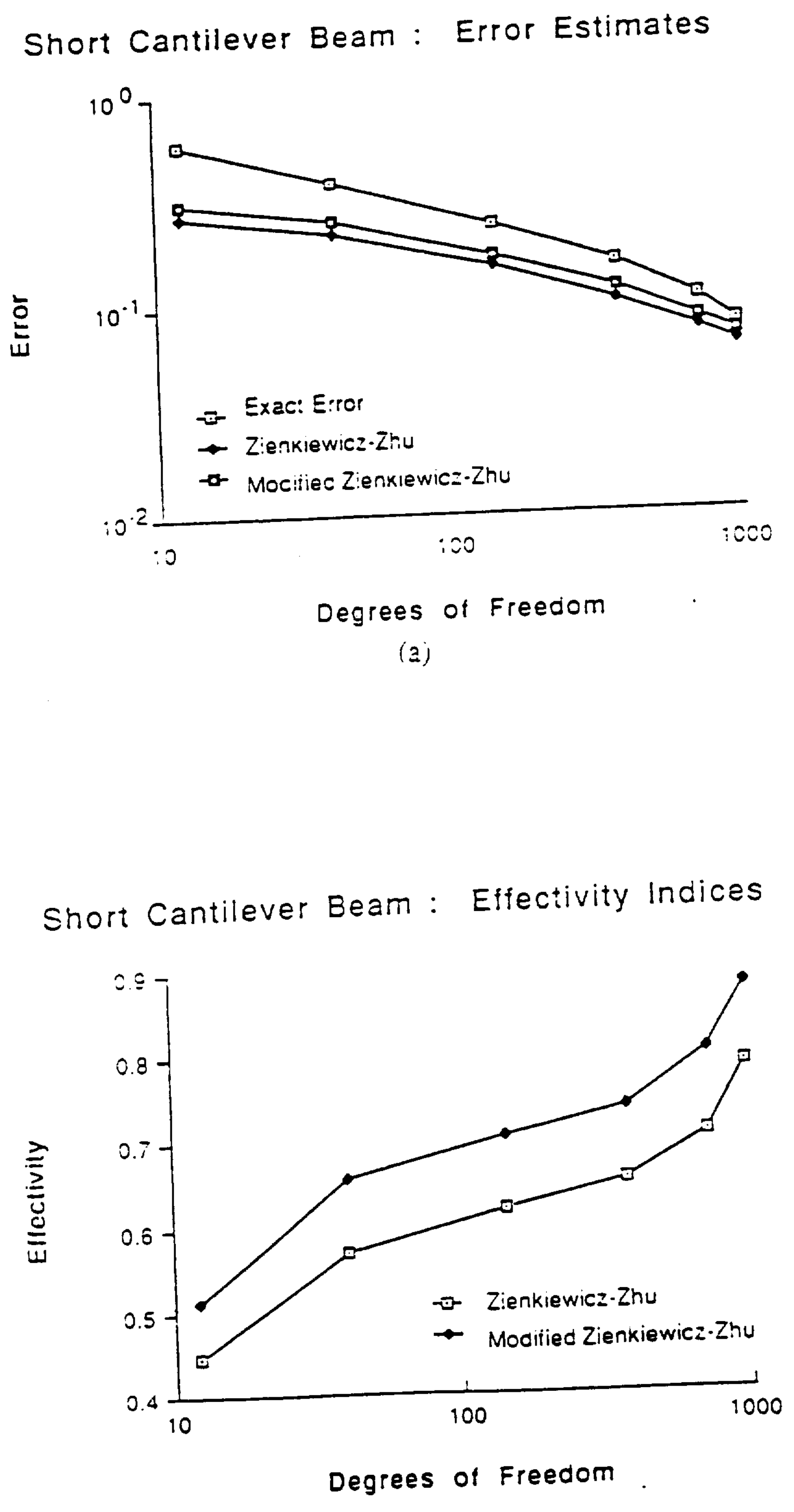
Let $\hat{X} \subset X$ denote a space of finite elements defined on a partitioning P of Ω into the union of subdomains $\{\Omega_K\}_{K=1}^N$ satisfying the usual assumptions. The finite element approximation $\hat{u} \approx u$ is defined by

$$\hat{u} \in \hat{X} : B(\hat{u}, \hat{v}) = L(\hat{v}) \quad \forall \hat{v} \in \hat{X} \qquad (213)$$

Having computed $\hat{u}$, it is usual to postprocess $\hat{u}$ to obtain some approximation to the gradient which we shall denote by $\hat{\boldsymbol{p}} \approx \nabla u$.

In the following sections, we consider various schemes by which $\hat{\boldsymbol{p}}$ can be obtained. For example, one may perform some projection of $\nabla\hat{u}$ onto smooth functions (see Hinton and Campbell [35], Rachowicz and Oden [36]), or, at the opposite extreme, simply take $\hat{\boldsymbol{p}} \approx \nabla\hat{u}$.

The study described in these sections attempts to answer the following questions.



(a)

(b)

Figure 3.67: (a) Exact error and two error estimates for the six meshes of the cantilever beam problem. (b) Effectivity indices of the two estimates.

- how accurate is the approximation $\hat{\boldsymbol{p}} \approx \boldsymbol{\nabla} u$ and can we estimate the accuracy numerically?

- how accurate is the approximation $\hat{\boldsymbol{p}} \approx \boldsymbol{\nabla} u$ *locally* and can we perform some local procedure by which to enhance the local approximation properties?

The first of these questions really addresses a much wider issue than is immediately apparent. Namely, the problem of designing a general purpose error estimation package and solution enhancement procedure.

The basic purpose of such a package is to accept some approximation $\hat{\boldsymbol{p}}$ to the true gradient of the solution of a given boundary value problem and to produce a numerical estimate of the accuracy $\hat{\boldsymbol{p}} \approx \boldsymbol{\nabla} u$ and, possibly, an improved approximation over $\hat{\boldsymbol{p}}$. Therefore, *the first question is essentially how to design an error estimation and postprocessing package.*

The second question recognizes that while many postprocessing schemes appear satisfactory when viewed at a global level, they may, nevertheless, perform poorly at a local level especially near boundaries of the domain or near singularities in the true solution. The question then arises of whether and how the approximation $\hat{\boldsymbol{p}} \approx \boldsymbol{\nabla} u$ may be enhanced in these areas.

A General Framework for Solution Enhancement and Accuracy Assessment

In this section we outline how one can mathematically formulate the problem discussed in the introduction. Suppose, once again, that $\hat{\boldsymbol{p}} \approx \boldsymbol{\nabla} u$ is some postprocessed gradient. Let the error in this approximation be denoted by $\boldsymbol{E}$. That is,

$$\boldsymbol{E} = \boldsymbol{\nabla} u - \hat{\boldsymbol{p}} \tag{214}$$

Obviously, we cannot compute $\boldsymbol{E}$ exactly. However, it is worthwhile to note that if we could compute $\boldsymbol{E}$ (or at least a good approximation to $\boldsymbol{E}$) then we not only have a means of computing the accuracy of $\hat{\boldsymbol{p}} \approx \boldsymbol{\nabla} u$, but also a means of improving $\hat{\boldsymbol{p}}$, thereby obtaining an enhanced approximation. That is, if $\boldsymbol{E}^* \approx \boldsymbol{E}$ then

$$\boldsymbol{\nabla} u = \hat{\boldsymbol{p}} + \boldsymbol{E} \approx \hat{\boldsymbol{p}} + \boldsymbol{E}^* \tag{215}$$

and, consequently, $\hat{\boldsymbol{p}} + \boldsymbol{E}^*$ is an enhanced approximation.

The question now, is how can $\boldsymbol{E}$ be approximated since $\boldsymbol{\nabla} u$ is unknown?

Let $v \in X$ (where X is given by (212)). Then, multiplying (214) by $\boldsymbol{\nabla} u$ and integrating over Ω gives

$$\int_\Omega \boldsymbol{E} \cdot \boldsymbol{\nabla} v \, d\boldsymbol{x} = \int_\Omega \boldsymbol{\nabla} u \cdot \boldsymbol{\nabla} v \, d\boldsymbol{x} - \int_\Omega \hat{\boldsymbol{p}} \cdot \boldsymbol{\nabla} v \, d\boldsymbol{x} \tag{216}$$

Now, recalling (211) and (212), we obtain

$$\begin{aligned}\int_\Omega \boldsymbol{E}\cdot\nabla v\;d\boldsymbol{x} &= B(u,v)-\int_\Omega \hat{\boldsymbol{p}}\cdot\nabla v\;d\boldsymbol{x}\\ &= L(v)-\int_\Omega \hat{\boldsymbol{p}}\cdot\nabla v\;d\boldsymbol{x}\\ &= \int_\Omega fv\;d\boldsymbol{x}+\int_{\Gamma_N} gv\;ds-\int_\Omega \hat{\boldsymbol{p}}\cdot\nabla v\;d\boldsymbol{x}\end{aligned}$$

That is, for any $v \in X$

$$\int_\Omega \boldsymbol{E}\cdot\nabla v\;d\boldsymbol{x} = \int_\Omega fv\;d\boldsymbol{x}+\int_{\Gamma_N} gv\;d\;s-\int_\Omega \hat{\boldsymbol{p}}\cdot\nabla v\;d\boldsymbol{x} \tag{217}$$

Examining (217), it is clear that every term on the right hand side can be computed. Therefore, it is possible to calculate $\boldsymbol{E}$ using (217) and hence obtain accuracy estimates and enhanced approximation as described above. While this approach is clear and straightforward, it does suffer an important drawback: it requires the approximation and the solution of a global problem and is therefore *expensive*.

One issue which we shall investigate is that of how to decompose (217) into a *sequence of local problems* on each element in the partition P. Solving a sequence of local problems is significantly less expensive than solving a single global problem.

Analysis of Nodal Averaging Postprocessing Schemes

We now take up the problem described in the previous section for the specific case of $\hat{\boldsymbol{p}}$ being a simple nodal average of gradients. For simplicity, we assume Ω (the domain) is rectangular and has been partitioned into the union of squares of side h. The space $\hat{X}$ is taken to consist of piecewise bilinear finite elements. Figure 3.68 shows how $\hat{\boldsymbol{p}}$ is defined at the nodes in the partition: for an interior node $\boldsymbol{x}_A$, we define

$$\hat{\boldsymbol{p}}(\boldsymbol{x}_A) = \frac{1}{4}\{\nabla\hat{u}|_{\boldsymbol{c}_1}+\nabla\hat{u}|_{\boldsymbol{c}_2}+\nabla\hat{u}|_{\boldsymbol{c}_3}+\nabla\hat{u}|_{\boldsymbol{c}_4}\} \tag{218}$$

where $\nabla\hat{u}|_{\boldsymbol{c}_1}$ denotes the direct approximation to the gradient evaluated at the centroid $\boldsymbol{c}_1$ of element Ω_1. For a boundary node $\boldsymbol{x}_B$ we define

$$\hat{\boldsymbol{p}}(\boldsymbol{x}_B) = \frac{1}{2}\{\nabla\hat{u}|_{\boldsymbol{c}_5}+\nabla\hat{u}|_{\boldsymbol{c}_6}\} \tag{219}$$

where Ω_5 and Ω_6 are shown in Fig. 3.68.

Proceeding in this manner allows us to obtain postprocessed values at each node in the mesh. Having done this, the value of $\hat{\boldsymbol{p}}$ at other points is obtained by means of bilinear interpolation of the nodal values.

This method is not being proposed as a postprocessing technique although it does coincide with a recognized superconvergent scheme on the interior of Ω (see Ainsworth and Craig

[15], Zlamal [46]). Since it represents a sensible choice for $\hat{\boldsymbol{p}}$, however, it has been chosen to illustrate our ideas. Furthermore, since this scheme is inferior (non-superconvergent) near the boundary, it represents a good choice for testing whether our methods for postprocessing can correctly identify the poor approximate gradients near the boundary and correct them accordingly.

Example: 1

As a first test problem we consider

$$\begin{aligned} -\nabla u &= 0 \text{ in } \Omega \\ u &= x^3 - 3xy^2 \text{ on } \Gamma_D \\ \frac{\partial u}{\partial n} &= \boldsymbol{n} \cdot \begin{pmatrix} 3x^2 - 3y^2 \\ -6xy \end{pmatrix} \text{ on } \Gamma_N \end{aligned} .$$

where Ω, Γ_N and Γ_D are shown in Figure 3.69. The true solution to the problem is $u = x^3 - 3xy^2$, meaning that we can compare our numerical approximations with the true solution.

In order to compare our subsequent local schemes with the accuracy attainable without localization, we first solve (217) globally, using the so-called Cantin-Loubignac Iteration, which we now briefly describe. First, we replace $\boldsymbol{E}$ by $\nabla\phi$ in (217) (where $\phi \in X$), giving

$$\phi \in X : \int_\Omega \nabla\phi \cdot \nabla v \; d\boldsymbol{x} = \int_\Omega fv \; d\boldsymbol{x} + \int_{\Gamma_N} gvds - \int_\Omega \hat{\boldsymbol{p}} \cdot \nabla v \; d\boldsymbol{x} \quad \forall v \in X \qquad (220)$$

where $\hat{\boldsymbol{p}}$ is the gradient obtained as described above. Integrating by parts allows us to write (220) in the form presented by Cantin and Loubignac originally:

$$\phi \in X : \int_\Omega \nabla\phi \cdot \nabla v \; d\boldsymbol{x} = \int_\Omega (f + \nabla \cdot \hat{\boldsymbol{p}})v \; d\boldsymbol{x} + \int_{\Gamma_N} (g - \boldsymbol{n} \cdot \hat{\boldsymbol{p}})vds \quad \forall v \in X \qquad (221)$$

We now discretize (221) in the same way as the original problem was discretized:

$$\hat{\phi} \in \hat{\boldsymbol{X}} : \int_\Omega \boldsymbol{\nabla}\hat{\phi} \cdot \boldsymbol{\nabla}\hat{v} \, d\boldsymbol{x} = \int_\Omega (f + \boldsymbol{\nabla} \cdot \hat{\boldsymbol{p}})\hat{v} \, d\boldsymbol{x} + \int_{\Gamma_N} (g - \boldsymbol{n} \cdot \hat{\boldsymbol{p}})\hat{v} \, ds \quad \forall \hat{v} \in \hat{X} \tag{222}$$

This problem is then solved in the same way as was $\hat{u}$. Having obtained $\hat{\phi}$ we set $\hat{\boldsymbol{E}} = \boldsymbol{\nabla}\hat{\phi} \approx \boldsymbol{E}$ and perform the correction

$$\left.\begin{aligned} \hat{u}^{\text{new}} &= \hat{u} + \hat{\phi} \\ \hat{\boldsymbol{p}}^{\text{new}} &= \boldsymbol{G}(\hat{u}^{\text{new}}) \end{aligned}\right\} \tag{223}$$

where $\boldsymbol{G}(\hat{u}^{\text{ new}})$ denotes the gradient approximation obtained by performing nodal averages of $\hat{u}^{\text{ new}}$. Naturally, we can now use $\hat{\boldsymbol{p}}^{\text{ new}}$ in place of $\hat{\boldsymbol{p}}$ in (222) and repeat the procedure. Table 4 shows the accuracy of these iterates along with the accuracy of $\boldsymbol{\nabla}\hat{u} \approx \boldsymbol{\nabla}u$ and $\hat{\boldsymbol{p}} \approx \boldsymbol{\nabla}u$ (i.e. when no correction has been performed). The following features can be seen:

- the uncorrected approximation $\hat{\boldsymbol{p}} \approx \boldsymbol{\nabla}u$ is inferior to not applying any smoothing. That is

  $$||\boldsymbol{\nabla}u - \boldsymbol{\nabla}\hat{u}||_{L^2(\Omega)} < ||\boldsymbol{\nabla}u - \hat{\boldsymbol{p}}||_{L^2(\Omega)}$$

  The reason for this can be seen by examining Table 5 which shows the contributions to the global error from each element. The effect of the nodal averaging is to degrade the approximation in elements adjacent to the boundary.

- the sequence of Cantin-Loubignac iterates provided a sequence of improved approximations to $\boldsymbol{\nabla}u$. These iterates converge in only a few steps and the greatest improvement is seen in the first application of the procedure.

- examining the local behavior of the iterates (Table 5), we see that the method correctly identifies the poor approximation of $\hat{\boldsymbol{p}}$ along the boundaries and improves the accuracy there.

In summary, the Cantin-Loubignac scheme works successfully, although the actual results are degraded owing to the deficiencies in the nodal recovery scheme being employed. Finally, we might ask what is the rate of convergence of the Cantin-Loubignac iterates as the partition is refined? Table 6 shows the results obtained on progressively finer meshes, from which it appears that the rate of convergence is slightly better than $\sigma(h)$.

Localization Strategies

Let us now return to our earlier question, namely that of how to localize the problem (217). Proceeding from (220) we have

$$\phi \in X : \int_\Omega \nabla\phi \cdot \nabla v \, d\boldsymbol{x} = \int_\Omega fv \, d\boldsymbol{x} + \int_{\Gamma_N} gvds - \int_\Omega \hat{\boldsymbol{p}} \cdot \nabla v \, d\boldsymbol{x} \quad \forall v \in X \tag{224}$$

To directly localize (224) will result in a very poor scheme. Therefore, we first rewrite (224) by integrating by parts and then decomposing into a sequence of problems over each element in the partition: for any $v \in X$,

$$\begin{aligned}
&\int_\Omega fv \, d\boldsymbol{x} + \int_{\Gamma_N} gvds - \int_\Omega \hat{\boldsymbol{p}} \cdot \nabla v \, d\boldsymbol{x} \\
&\qquad = \int_\Omega (f + \nabla \cdot \hat{\boldsymbol{p}}) v \, d\boldsymbol{x} + \int_{\Gamma_N} (g - \boldsymbol{n} \cdot \hat{\boldsymbol{p}}) vds \\
&\qquad = \sum_{K=1}^{N} \left\{ \int_\Omega (f + \nabla \cdot \hat{\boldsymbol{p}}) v \, d\boldsymbol{x} + \int_{\partial\Omega_K \bigcap \Gamma_N} (g - \boldsymbol{n} \cdot \hat{\boldsymbol{p}}) vds \right\} \\
&\qquad = \sum_{K=1}^{N} \left\{ \int_{\Omega_K} fv \, d\boldsymbol{x} + \int_{\partial\Omega_K \bigcap \Gamma_N} gvds - \int_{\Omega_K} \hat{\boldsymbol{p}} \cdot \nabla v \, d\boldsymbol{x} \right. \\
&\qquad + \left. \int_{\partial\Omega_K \setminus \Gamma_N} (\boldsymbol{n}_K \cdot \hat{\boldsymbol{p}}) vds \right\}
\end{aligned} \tag{225}$$

In view of (225), we decompose (224) into the sequence of local problems:

$$\begin{aligned}
\phi_K \in X_K : \int_{\Omega_K} \nabla\phi_K \cdot \nabla v \, d\boldsymbol{x} &= \textstyle\int_{\Omega_K} fv \, d\boldsymbol{x} + \displaystyle\int_{\partial\Omega_K \bigcap \Gamma_N} gvds \\
&\quad - \textstyle\int_{\Omega_K} \hat{\boldsymbol{p}} \cdot \nabla v d\boldsymbol{x} + \int_{\partial\Omega_K \setminus \Gamma_N} \boldsymbol{n}_K \cdot \hat{\boldsymbol{p}} vds \\
&\qquad \forall v \in X_K
\end{aligned} \tag{226}$$

where $X_K = \{v \in H^1(\Omega_K) : \gamma v = 0 \text{ on } \partial\Omega_K \bigcap \Gamma_D\}$ and $\boldsymbol{n}_K$ denotes the unit outward normal vector on $\partial\Omega_K$.

One advantage of the localization process is that we can calculate a high order approximation $\hat{\phi}_K$ to the solution of this problem. That is, we need not restrict ourselves to calculating a bilinear approximation. However, there are drawbacks to the localization. In particular, the boundary conditions for the local problem are imprecise, whereas they are known exactly for the global problem. Therefore, the major difficulty in localizing the problem is the determination of the boundary conditions on edges which lie on the interior of the mesh.

Example: 2

The purpose of this example is to investigate the influence of various choices of boundary conditions on the interior edges. Once again we use the simple test problem

$$\left.\begin{aligned} -\nabla u = 0 \quad &\text{in} \quad \Omega \\ u = \bar{u} \quad &\text{on} \quad \Gamma_D \\ \frac{\partial u}{\partial n} = g \quad &\text{on} \quad \Gamma_N \end{aligned}\right\} \tag{227}$$

where Γ_D, Γ_N are shown in Figure 3.69. As before the true solution is $u(x,y) = x^3 - 3xy^2$ in Ω. In Table 7 we compare the accuracy obtained by solving the local problems with three choices of boundary conditions. A typical local problem stated in strong form is

Find ϕ_K:

$$\left.\begin{aligned} -\nabla\phi_K &= f + \boldsymbol{\nabla}\cdot\hat{\boldsymbol{p}} && \text{in } \Omega_K \\ \frac{\partial\phi_K}{\partial n} &= g - \boldsymbol{n}\cdot\hat{\boldsymbol{p}} && \text{on } \partial\Omega_K \bigcap \Gamma_N \\ \phi_K &= \bar{u} - \hat{u} && \text{on } \partial\Omega_K \bigcap \Gamma_D \\ \frac{\partial\phi_K}{\partial n_K} &= g^* - \boldsymbol{n}_K\cdot\hat{\boldsymbol{p}} && \text{on } \partial\Omega_K \backslash \Gamma \end{aligned}\right\} \tag{228}$$

where g^* should be chosen as an approximation to the true flux $\boldsymbol{n}_K \cdot \boldsymbol{\nabla} u$. We show results for the following choices:

(a) $g^* = \frac{\partial u}{\partial n_K}$ (i.e. using true flux)

(b) $g^* = \boldsymbol{n}_K \cdot \hat{\boldsymbol{p}}$ (i.e. using approximation to true flux based on nodal averaging)

(c) $g^* = \frac{1}{2}\boldsymbol{n}_K \cdot (\boldsymbol{\nabla}\hat{u}|_{\text{left}} + \boldsymbol{\nabla}\hat{u}|_{\text{right}}) = \langle\frac{\partial\hat{u}}{\partial n_K}\rangle$ (i.e. the flux approximation obtained by averaging fluxes between neighboring elements)

The first case is not a practical method, but is included to show the best results which can be attained. Examination of Table 7 shows that choice (b) performs rather poorly due to the behavior in elements along the Dirichlet boundary. Better results are obtained using choice (c) which is almost as good as using the true fluxes

A more significant conclusion is reached by comparing the accuracy of the local scheme with the full Cantin-Loubignac iteration. Referring to Table 4, one sees that at convergence the accuracy of the full Cantin-Loubignac scheme is .298 compared with .201 obtained using the local scheme. The cost of the local scheme is considerably less than the full scheme and is more accurate.

The conclusions which we may draw from this simple investigation are:

1. *not only is a localized version of a Cantin-Loubignac scheme viable, but in fact, performs better than the global scheme* (owing to the possibility of solving local problems using high order methods).

2. *the choice of boundary condition for the local problems is critical.*

To enlarge upon the second point, we have seen that neither the Cantin-Loubignac scheme or the localized version can fully compensate for the deficiencies in a poor choice of postprocessing scheme to obtain $\hat{\boldsymbol{p}}$. However, they can locate the areas of the mesh where the scheme is poor.

Further Numerical Studies

In this section we continue the study outlined previously. However, we now consider an alternative choice of postprocessing by which $\hat{\boldsymbol{p}}$ can be obtained. Previously, we employed a deliberately poor scheme.

The choice of $\hat{\boldsymbol{p}}$ is similar to before, except that we now do not choose centroid values (c.f 218-219) but choose to average values at the node. That is (c.f. Figure 3.68)

$$\hat{\boldsymbol{p}}(\boldsymbol{x}_A) = \frac{1}{4}\{\boldsymbol{\nabla}\hat{u}|_{\boldsymbol{x}_A,\Omega_1} + \boldsymbol{\nabla}\hat{u}|_{\boldsymbol{x}_A,\Omega_2} + \boldsymbol{\nabla}\hat{u}|_{\boldsymbol{x}_A,\Omega_3} + \boldsymbol{\nabla}\hat{u}|_{\boldsymbol{x}_A,\Omega_4}\} \tag{229}$$

where $\boldsymbol{\nabla}\hat{u}|_{\boldsymbol{x}_A,\Omega_1}$ denotes the value of $\boldsymbol{\nabla}\hat{u}$ obtained by evaluating $\boldsymbol{\nabla}\hat{u}$ on element Ω_1 at global coordinate $\boldsymbol{x}_A$. The values at other nodes are determined in a similar way. Values at other points are obtained by bilinear interpolation of the nodal values. This scheme is *not* superconvergent, but once again represents a sensible choice of postprocessing.

Example: 3: Full Cantin-Loubignac Scheme

Once again we solve the simple problem of the previous examples, but this time we use the full Cantin-Loubignac scheme in conjunction with the recovery scheme described above. The accuracy of the Cantin-Loubignac iterates are shown in Tables 8 and 9.

Comparing these results with Tables 4-5, it is seen that the new choice of $\hat{\boldsymbol{p}}$ leads to a significant improvement in the accuracy of the postprocessed gradient and the iterates.

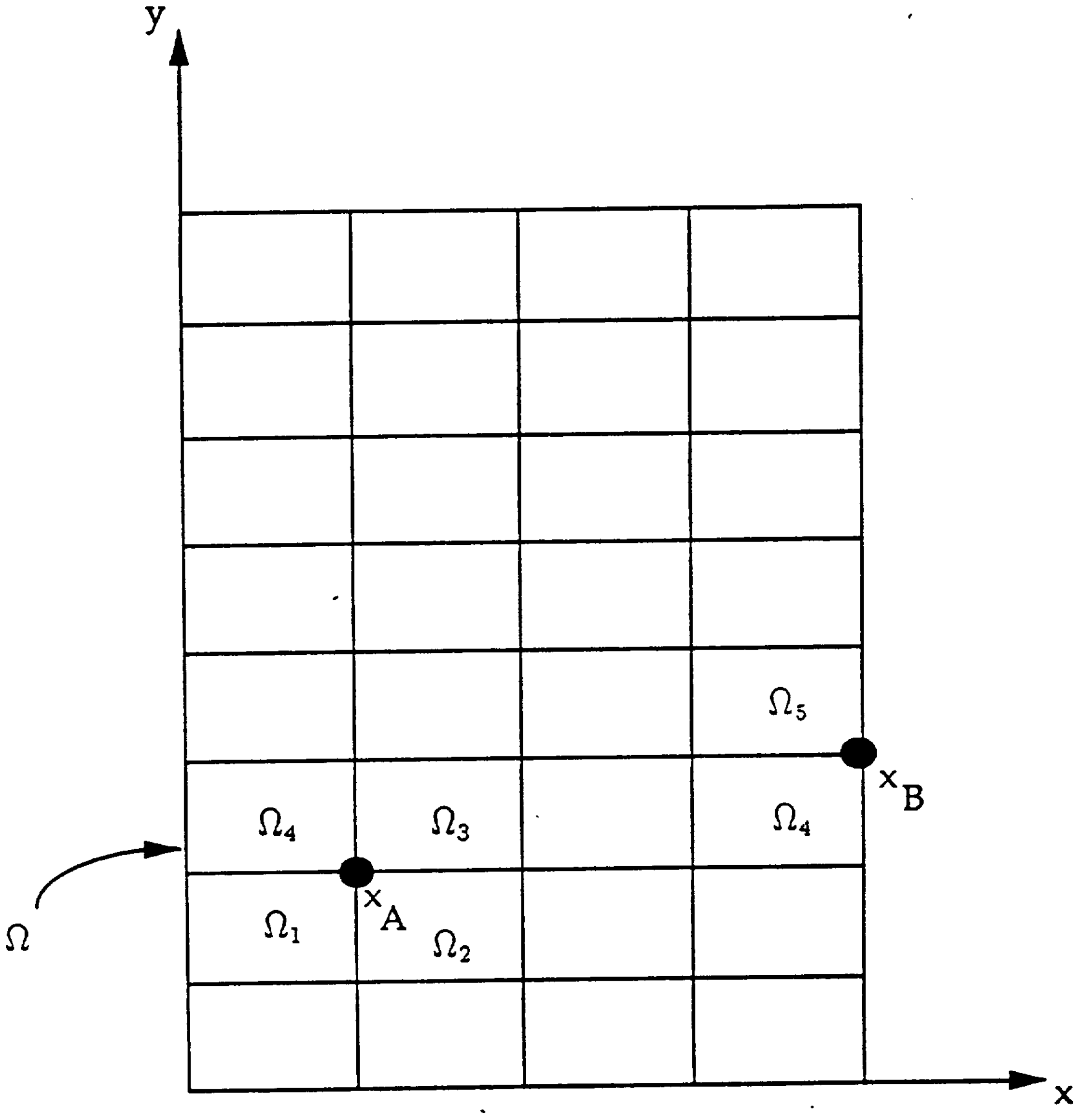


Figure 3.68: Postprocessing by nodal averaging.

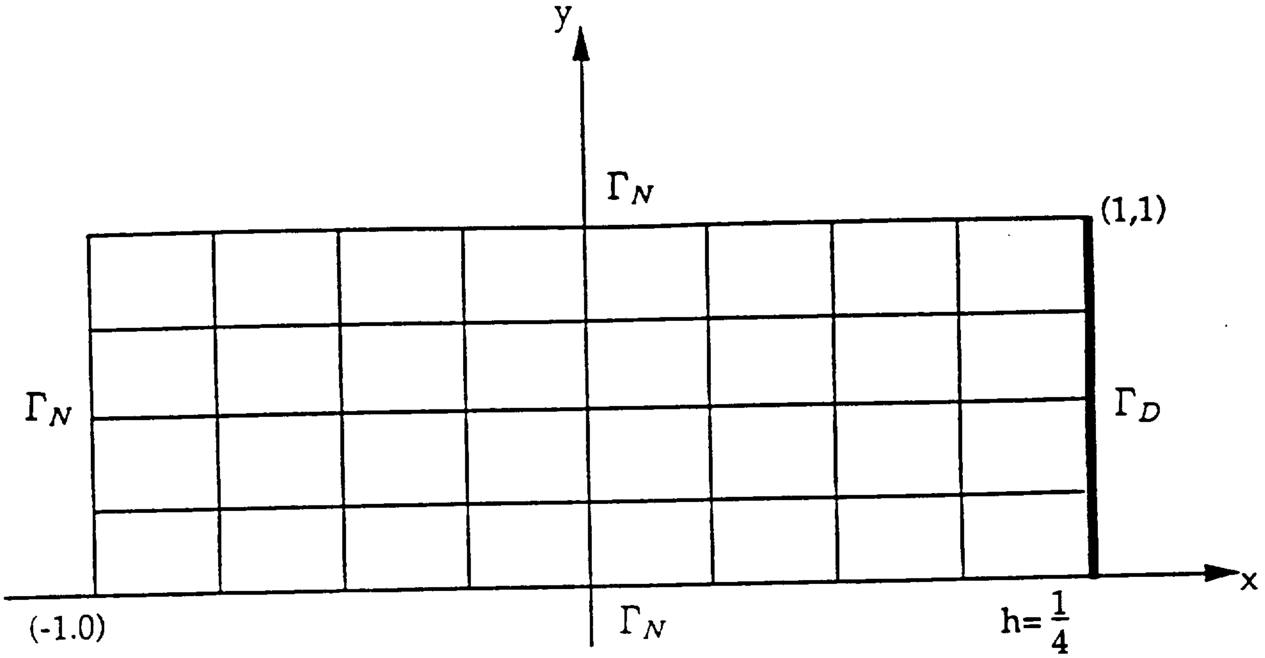


Figure 3.69: Domain, mesh and boundary conditions for the first example.

Iterate	$\|\|\nabla u - \hat{p}\|\|_{L^2(\Omega)}$
0	.515909
1	.363061
2	.336462
3	.324060
4	.316089
5	.310410
6	.306194
7	.303001
8	.300554
9	.298661

$||e||_E = .498827$

Table 4: Accuracy of Cantin-Loubignac iterates for the first example.

ERRORS IN DIRECT GRADIENT APPROXIMATION:
Error in H^1: 0.49882675632465

Row: 4	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01
Row: 3	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01
Row: 2	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01
Row: 1	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01

ERRORS IN NODALLY AVERAGED GRADIENT: **Iterate 0**
Error in H^1: 0.51590918874375

Row: 4	0.361E-01	0.164E-01	0.135E-01	0.120E-01	0.120E-01	0.135E-01	0.164E-01	0.361E-01
Row: 3	0.164E-01	0.244E-04	0.244E-04	0.244E-04	0.244E-04	0.244E-04	0.244E-04	0.164E-01
Row: 2	0.135E-01	0.244E-04	0.244E-04	0.244E-04	0.244E-04	0.244E-04	0.244E-04	0.135E-01
Row: 1	0.185E-01	0.466E-02	0.173E-02	0.269E-03	0.269E-03	0.173E-02	0.466E-02	0.185E-01

ERRORS IN NODALLY AVERAGED GRADIENT: **Iterate 1**
Error in H^1: 0.36306111271486

Row: 4	0.786E-02	0.760E-02	0.589E-02	0.658E-02	0.694E-02	0.720E-02	0.129E-01	0.274E-01
Row: 3	0.547E-02	0.257E-02	0.118E-02	0.626E-03	0.516E-03	0.622E-03	0.419E-03	0.777E-02
Row: 2	0.317E-02	0.716E-03	0.303E-03	0.365E-03	0.420E-03	0.493E-03	0.281E-03	0.701E-02
Row: 1	0.417E-02	0.193E-02	0.711E-03	0.538E-03	0.616E-03	0.915E-03	0.167E-02	0.699E-02

ERRORS IN NODALLY AVERAGED GRADIENT: **Iterate 2**
Error in H^1: 0.33646193862482

Row: 4	0.519E-02	0.652E-02	0.528E-02	0.668E-02	0.697E-02	0.617E-02	0.111E-01	0.247E-01
Row: 3	0.409E-02	0.302E-02	0.149E-02	0.608E-03	0.508E-03	0.923E-03	0.657E-03	0.600E-02
Row: 2	0.215E-02	0.837E-03	0.515E-03	0.376E-03	0.424E-03	0.738E-03	0.428E-03	0.577E-02
Row: 1	0.228E-02	0.127E-02	0.674E-03	0.520E-03	0.622E-03	0.108E-02	0.110E-02	0.448E-02

ERRORS IN NODALLY AVERAGED GRADIENT: **Iterate 3**
Error in H^1: 0.32406025288887

Row: 4	0.425E-02	0.592E-02	0.531E-02	0.696E-02	0.727E-02	0.596E-02	0.100E-01	0.231E-01
Row: 3	0.335E-02	0.298E-02	0.158E-02	0.556E-03	0.459E-03	0.994E-03	0.737E-03	0.531E-02
Row: 2	0.179E-02	0.832E-03	0.632E-03	0.343E-03	0.383E-03	0.803E-03	0.484E-03	0.519E-02
Row: 1	0.167E-02	0.999E-03	0.671E-03	0.437E-03	0.532E-03	0.113E-02	0.929E-03	0.345E-02

Table 5: Element by element errors for iterates in Cantin Loubignac Scheme for the first example.

Iterate	$\|\|\nabla u - \hat{\boldsymbol{p}}\|\|_{L^2(\Omega)}$
0	.186232
1	.132681
2	.124869
3	.121490
4	.119525
5	.118210
6	.117248
7	.116498
8	.115886

$||\nabla u - \nabla \hat{u}||_{L^2(\Omega)}= .249853$
$h= \frac{1}{8}$

Iterate	$\|\|\nabla u - \hat{\boldsymbol{p}}\|\|_{L^2(\Omega)}$
0	.066187
1	.048201
2	.045755
3	.044704
4	.044098
5	.043694
6	.043401
7	.043176
8	.042997

$||\nabla u - \nabla \hat{u}||_{L^2(\Omega)}= .124982$
$h= \frac{1}{16}$

Table 6: Accuracy of Cantin-Loubignac iterates for the first example on finer meshes.

ERRORS IN DIRECT GRADIENT APPROXIMATION:
Error in H^1:   0.49882675632465

Row: 4	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01
Row: 3	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01
Row: 2	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01
Row: 1	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01

ERRORS IN CORRECTED GRADIENT:
Error in H^1:   0.17647157085140

Row: 4	0.540E-05	0.949E-03	0.949E-03	0.949E-03	0.949E-03	0.949E-03	0.949E-03	0.174E-01
Row: 3	0.485E-03	0.911E-05	0.911E-05	0.911E-05	0.911E-05	0.911E-05	0.911E-05	0.366E-02
Row: 2	0.176E-03	0.911E-05	0.911E-05	0.911E-05	0.911E-05	0.911E-05	0.911E-05	0.208E-02
Row: 1	0.540E-05	0.216E-04	0.216E-04	0.216E-04	0.216E-04	0.216E-04	0.216E-04	0.136E-02

(a) Results obtained with exact boundary conditions.

ERRORS IN CORRECTED GRADIENT:
Error in H^1:   0.40907901358811

Row: 4	0.465E-01	0.100E-01	0.100E-01	0.100E-01	0.100E-01	0.100E-01	0.100E-01	0.314E-0
Row: 3	0.917E-02	0.732E-04	0.732E-04	0.732E-04	0.732E-04	0.732E-04	0.732E-04	0.610E-0
Row: 2	0.514E-02	0.732E-04	0.732E-04	0.732E-04	0.732E-04	0.732E-04	0.732E-04	0.317E-0
Row: 1	0.256E-02	0.857E-04	0.857E-04	0.857E-04	0.857E-04	0.857E-04	0.857E-04	0.177E-0

(b) Results obtained using $\boldsymbol{n} \cdot \hat{\boldsymbol{p}}$ as a boundary condition.

ERRORS IN CORRECTED GRADIENT:
Error in H^1:   0.20186638019028

Row: 4	0.678E-03	0.120E-02	0.120E-02	0.120E-02	0.120E-02	0.120E-02	0.120E-02	0.177E-01
Row: 3	0.116E-02	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.390E-02
Row: 2	0.849E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.232E-02
Row: 1	0.678E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.160E-02

(c) Results obtained using $\langle \frac{\partial \hat{u}}{\partial n} \rangle$ as a boundary condition.

Table 7: Impact of different choices of boundary condition for localized problems.

Iterate	$\|\|\nabla u - \hat{\boldsymbol{p}}\|\|_{L^2(\Omega)}$
0	.373434
1	.220942
2	.190204
3	.174521
4	.164552
5	.157789
6	.153111
7	.149872
8	.147641
9	.146119

Table 8: Accuracy of Cantin-Loubignac iterates for the example problem.

ERRORS IN DIRECT GRADIENT APPROXIMATION:
Error in H^1: 0.49882675632465

Row: 4	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01
Row: 3	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01
Row: 2	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01
Row: 1	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01

ERRORS IN NODALLY AVERAGED GRADIENT: **Iterate 0**
Error in H^1: 0.37343449828838

Row: 4	0.169E-01	0.521E-02	0.228E-02	0.818E-03	0.818E-03	0.228E-02	0.521E-02	0.169E-01
Row: 3	0.790E-02	0.574E-03	0.574E-03	0.574E-03	0.574E-03	0.574E-03	0.574E-03	0.790E-02
Row: 2	0.790E-02	0.574E-03	0.574E-03	0.574E-03	0.574E-03	0.574E-03	0.574E-03	0.790E-02
Row: 1	0.169E-01	0.521E-02	0.228E-02	0.818E-03	0.818E-03	0.228E-02	0.521E-02	0.169E-01

ERRORS IN NODALLY AVERAGED GRADIENT: **Iterate 1**
Error in H^1: 0.22094244582713

Row: 4	0.432E-02	0.147E-02	0.424E-03	0.150E-03	0.148E-03	0.427E-03	0.180E-02	0.736E-02
Row: 3	0.256E-02	0.129E-02	0.164E-03	0.737E-04	0.725E-04	0.152E-03	0.983E-03	0.311E-02
Row: 2	0.256E-02	0.129E-02	0.164E-03	0.737E-04	0.725E-04	0.152E-03	0.883E-03	0.311E-02
Row: 1	0.432E-02	0.147E-02	0.424E-03	0.150E-03	0.148E-03	0.427E-03	0.180E-02	0.736E-02

ERRORS IN NODALLY AVERAGED GRADIENT: **Iterate 2**
Error in H^1: 0.19020373084707

Row: 4	0.255E-02	0.105E-02	0.431E-03	0.154E-03	0.148E-03	0.449E-03	0.123E-02	0.604E-02
Row: 3	0.173E-02	0.109E-02	0.143E-03	0.756E-04	0.713E-04	0.131E-03	0.764E-03	0.203E-02
Row: 2	0.173E-02	0.109E-02	0.143E-03	0.756E-04	0.713E-04	0.131E-03	0.764E-03	0.203E-02
Row: 1	0.255E-02	0.105E-02	0.431E-03	0.154E-03	0.148E-03	0.449E-03	0.123E-02	0.604E-02

ERRORS IN NODALLY AVERAGED GRADIENT: **Iterate 3**
Error in H^1: 0.17452057639637

Row: 4	0.185E-02	0.859E-03	0.433E-03	0.142E-03	0.132E-03	0.464E-03	0.954E-03	0.556E-02
Row: 3	0.131E-02	0.859E-03	0.156E-03	0.910E-04	0.852E-04	0.144E-03	0.629E-03	0.156E-02
Row: 2	0.131E-02	0.859E-03	0.156E-03	0.910E-04	0.852E-04	0.144E-03	0.629E-03	0.156E-02
Row: 1	0.185E-02	0.859E-03	0.433E-03	0.142E-03	0.132E-03	0.464E-03	0.954E-03	0.556E-02

Table 9: Element by element errors for iterates in Cantin-Loubignac scheme for the example problem.

In Table 11 we repeat our earlier experiment concerning the various choices of boundary condition for the local problem. First, we note that *the localized schemes once again are superior to the full Cantin-Loubignac scheme.* Second, the choice of boundary condition is not so critical for the improved recovery scheme determining $\hat{\boldsymbol{p}}$. The approximate choice of boundary condition is almost as good as the true boundary condition. Furthermore, the local behavior of the schemes is satisfactory and identifies the poor approximation $\hat{\boldsymbol{p}} \approx \boldsymbol{\nabla} u$ in the neighborhood of the boundaries and corrects accordingly.

Example: Cracked Panel Problem

In the examples considered so far, the solution has been smooth. In order to consider the effect of singularities, we now study the case of a cracked panel problem:

Find u:

$$\begin{aligned} -\nabla u &= 0 \text{ in } \Omega \\ u &= \bar{u} \text{ on } \Gamma_D \\ \frac{\partial u}{\partial n} &= 0 \text{ on } \Gamma_N \end{aligned}$$

where Ω, Γ_D and Γ_N are shown in Figure 3.70. The true solution to this problem is

$$u(x,y) = r^{\frac{1}{2}} \cos \frac{1}{2} \theta$$

so that there is a singularity in the stresses near the origin (i.e. at the tip of the crack).

The results obtained for the full Cantin-Loubignac scheme are shown in Tables 12-13, from which one sees that the scheme has little success in correcting the poor approximation to the gradient in the neighborhood of the singularity.

The behavior of the localized scheme is shown in Table 14. Once again we pay attention to the effect of boundary conditions for the local problem with exact boundary conditions the localized scheme is superior to the full Cantin-Loubignac scheme. However, the results show how sensitive the method is to the boundary condition. The results obtained with the approximate choice of fluxes corresponding to $\boldsymbol{n}_K \cdot \hat{\boldsymbol{p}}$ proving completely unsatisfactory.

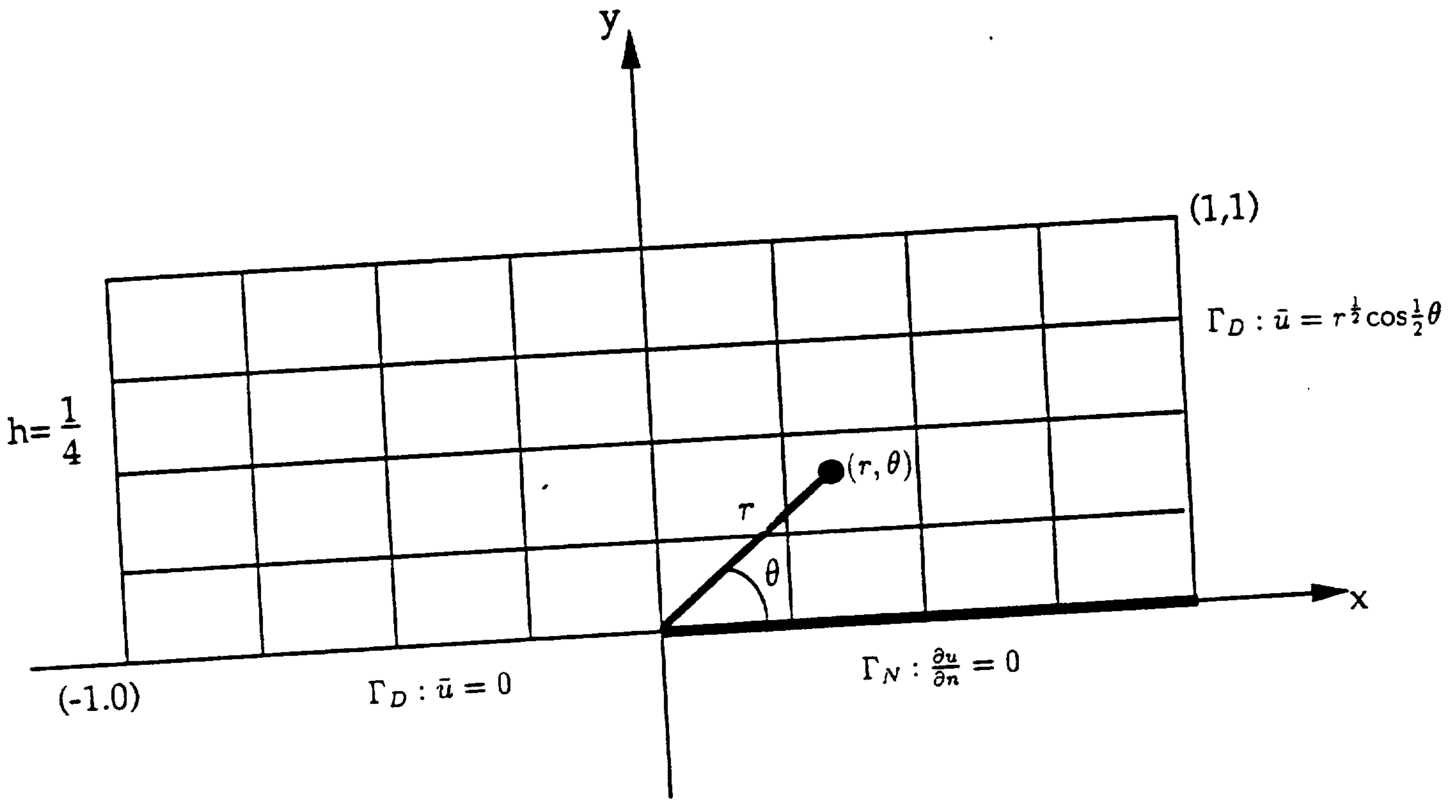

y
(1,1)
$\Gamma_D : \bar{u} = r^{\frac{1}{2}} \cos \frac{1}{2}\theta$
h= $\frac{1}{4}$
(r, θ)
r
θ
x
(-1.0)
$\Gamma_D : \bar{u} = 0$
$\Gamma_N : \frac{\partial u}{\partial n} = 0$

Figure 3.70: Domain, mesh and boundary conditions for Cracked Panel Problem.

Iterate	$\lVert\boldsymbol{\nabla} u - \hat{\boldsymbol{p}}\rVert_{L^2(\Omega)}$
0	.1359
1	7.416(-2)
2	6.208(-2)
3	5.652(-2)
4	5.321(-2)
5	5.097(-2)
6	4.932(-2)
7	4.802(-2)
8	4.670(-2)

$h = \frac{1}{8}$

Iterate	$\lVert\boldsymbol{\nabla} u - \hat{\boldsymbol{p}}\rVert_{L^2(\Omega)}$
0	4.874(-2)
1	2.548(-2)
2	2.101(-2)
3	1.890(-2)
4	1.761(-2)
5	1.673(-2)
6	1.607(-2)
7	1.556(-2)
8	1.513(-2)

$h = \frac{1}{16}$

Table 10: Accuracy of Cantin-Loubignac iterates for the smooth example on finer meshes.

ERRORS IN DIRECT GRADIENT APPROXIMATION:
Error in H^1:   0.49882675632465

Row: 4	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01
Row: 3	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01
Row: 2	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01
Row: 1	0.180E-01	0.924E-02	0.338E-02	0.452E-03	0.452E-03	0.338E-02	0.924E-02	0.180E-01

ERRORS IN CORRECTED GRADIENT:
Error in H^1:   9.1923280705362D-02

Row: 4	0.216E-04	0.216E-04	0.216E-04	0.216E-04	0.216E-04	0.216E-04	0.216E-04	0.250E-02
Row: 3	0.911E-05	0.911E-05	0.911E-05	0.911E-05	0.911E-05	0.911E-05	0.911E-05	0.151E-02
Row: 2	0.911E-05	0.911E-05	0.911E-05	0.911E-05	0.911E-05	0.911E-05	0.911E-05	0.151E-02
Row: 1	0.216E-04	0.216E-04	0.216E-04	0.216E-04	0.216E-04	0.216E-04	0.216E-04	0.250E-02

(a) Results obtained with exact boundary conditions.

ERRORS IN CORRECTED GRADIENT:
Error in H^1:   0.13437981713121

Row: 4	0.695E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.274E-02
Row: 3	0.682E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.176E-02
Row: 2	0.682E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.176E-02
Row: 1	0.695E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.274E-02

(b) Results obtained using n.p as a boundary condition.

ERRORS IN CORRECTED GRADIENT:
Error in H^1:   0.13437981713121

Row: 4	0.695E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.274E-02
Row: 3	0.682E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.176E-02
Row: 2	0.682E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.256E-03	0.176E-02
Row: 1	0.695E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.269E-03	0.274E-02

(c) Results obtained using $\langle \frac{du}{dn} \rangle$ as a boundary condition.

Table 11: Impact of different choices of boundary condition for localized problems.

Iterate	$\|\|\nabla u - \hat{p}\|\|_{L^2(\Omega)}$
0	.515909
1	.363061
2	.336462
3	.324060
4	.316089
5	.310410
6	.306194
7	.303001
8	.300554
9	.298661

$$||\nabla u - \nabla \hat{u}||_{L^2(\Omega)} = 0.200177$$

Table 12: Accuracy of Cantin-Loubignac iterates for the Cracked Panel example.

ERRORS IN DIRECT GRADIENT APPROXIMATION:

Error in H^1: 0.20017721413491

Row: 4	0.200E-04	0.356E-04	0.526E-04	0.569E-04	0.498E-04	0.434E-04	0.280E-04	0.117E-04
Row: 3	0.271E-04	0.617E-04	0.117E-03	0.150E-03	0.953E-04	0.833E-04	0.701E-04	0.365E-04
Row: 2	0.236E-04	0.766E-04	0.333E-03	0.649E-03	0.414E-03	0.528E-03	0.123E-03	0.724E-04
Row: 1	0.605E-05	0.333E-04	0.329E-03	0.187E-01	0.168E-01	0.734E-03	0.211E-03	0.101E-03

ERRORS IN NODALLY AVERAGED GRADIENT: Iterate 0

Error in H^1: 0.18481387268082

Row: 4	0.197E-04	0.343E-04	0.606E-04	0.839E-04	0.874E-04	0.557E-04	0.177E-04	0.957E-05
Row: 3	0.330E-05	0.540E-05	0.339E-04	0.809E-04	0.123E-03	0.672E-04	0.230E-04	0.488E-04
Row: 2	0.218E-05	0.570E-05	0.208E-04	0.637E-03	0.924E-03	0.497E-04	0.954E-04	0.114E-03
Row: 1	0.557E-05	0.383E-04	0.265E-03	0.110E-01	0.183E-01	0.144E-02	0.249E-03	0.195E-03

ERRORS IN NODALLY AVERAGED GRADIENT: Iterate 1

Error in H^1: 0.16801458744145

Row: 4	0.193E-04	0.109E-04	0.210E-04	0.321E-04	0.142E-04	0.286E-05	0.933E-05	0.627E-05
Row: 3	0.162E-04	0.203E-05	0.557E-05	0.151E-04	0.825E-05	0.515E-05	0.943E-05	0.268E-05
Row: 2	0.134E-04	0.577E-05	0.292E-04	0.542E-03	0.613E-03	0.217E-03	0.199E-04	0.698E-05
Row: 1	0.483E-05	0.677E-05	0.907E-04	0.102E-01	0.148E-01	0.145E-02	0.796E-04	0.146E-04

ERRORS IN NODALLY AVERAGED GRADIENT: Iterate 2

Error in H^1: 0.16596666832397

Row: 4	0.198E-04	0.883E-05	0.178E-04	0.305E-04	0.102E-04	0.132E-04	0.200E-04	0.887E-05
Row: 3	0.187E-04	0.274E-05	0.380E-05	0.128E-04	0.109E-04	0.199E-04	0.191E-04	0.561E-05
Row: 2	0.191E-04	0.147E-04	0.343E-04	0.484E-03	0.612E-03	0.363E-03	0.137E-04	0.903E-05
Row: 1	0.728E-05	0.113E-04	0.601E-04	0.101E-01	0.137E-01	0.169E-02	0.217E-03	0.981E-05

ERRORS IN NODALLY AVERAGED GRADIENT: Iterate 3

Error in H^1: 0.16576804732717

Row: 4	0.203E-04	0.848E-05	0.158E-04	0.306E-04	0.113E-04	0.217E-04	0.226E-04	0.911E-05
Row: 3	0.183E-04	0.307E-05	0.580E-05	0.184E-04	0.253E-04	0.342E-04	0.241E-04	0.693E-05
Row: 2	0.207E-04	0.221E-04	0.392E-04	0.450E-03	0.643E-03	0.447E-03	0.189E-04	0.118E-04
Row: 1	0.884E-05	0.184E-04	0.522E-04	0.102E-01	0.131E-01	0.185E-02	0.325E-03	0.139E-04

Table 13: Element by element errors for iterates in Cantin-Loubignac scheme for Cracked Panel.

Summary and Conclusions

The basic aim of the work described in this section was to see whether the extremely expensive full Cantin-Loubignac scheme could be localized (and therefore be made cheaper) without sacrificing too much accuracy. The results presented here show that the answer is affirmative for smooth problems. In fact, the localized scheme can be superior (owing to the higher order accuracy by which the local problems can be solved). However, the determination of boundary conditions for the local problems is critical. The simple schemes presented here work well for smooth problem but are unsatisfactory for a non-smooth case.

ERRORS IN DIRECT GRADIENT APPROXIMATION:
Error in H^1: 0.20017721413491

Row: 4	0.200E-04	0.356E-04	0.526E-04	0.569E-04	0.498E-04	0.434E-04	0.280E-04	0.117E-04
Row: 3	0.271E-04	0.617E-04	0.117E-03	0.150E-03	0.953E-04	0.833E-04	0.701E-04	0.365E-04
Row: 2	0.236E-04	0.766E-04	0.333E-03	0.649E-03	0.414E-03	0.528E-03	0.123E-03	0.724E-04
Row: 1	0.605E-05	0.333E-04	0.329E-03	0.187E-01	0.168E-01	0.734E-03	0.211E-03	0.101E-03

ERRORS IN CORRECTED GRADIENT:
Error in H^1: 0.11623575645905

Row: 4	0.103E-04	0.233E-04	0.442E-04	0.661E-04	0.291E-04	0.238E-04	0.118E-04	0.738E-05
Row: 3	0.187E-05	0.525E-05	0.334E-04	0.800E-04	0.958E-06	0.208E-05	0.848E-07	0.395E-04
Row: 2	0.210E-05	0.526E-05	0.164E-04	0.606E-03	0.152E-03	0.117E-05	0.351E-05	0.935E-04
Row: 1	0.560E-05	0.380E-04	0.258E-03	0.107E-01	0.973E-03	0.149E-03	0.374E-06	0.145E-03

(a) Results obtained using exact fluxes for boundary conditions.

ERRORS IN CORRECTED GRADIENT:
Error in H^1: 0.30233586712545

Row: 4	0.103E-04	0.233E-04	0.442E-04	0.661E-04	0.603E-04	0.444E-04	0.175E-04	0.738E-05
Row: 3	0.187E-05	0.525E-05	0.334E-04	0.800E-04	0.105E-03	0.102E-03	0.910E-04	0.395E-04
Row: 2	0.210E-05	0.526E-05	0.164E-04	0.606E-03	0.695E-03	0.169E-02	0.870E-04	0.935E-04
Row: 1	0.560E-05	0.380E-04	0.258E-03	0.107E-01	0.746E-01	0.157E-02	0.141E-03	0.145E-03

(b) Results obtained using n.p as a boundary condition.

ERRORS IN CORRECTED GRADIENT:
Error in H^1: 0.30233586712545

Row: 4	0.103E-04	0.233E-04	0.442E-04	0.661E-04	0.603E-04	0.444E-04	0.175E-04	0.738E-05
Row: 3	0.187E-05	0.525E-05	0.334E-04	0.800E-04	0.105E-03	0.102E-03	0.910E-04	0.395E-04
Row: 2	0.210E-05	0.526E-05	0.164E-04	0.606E-03	0.695E-03	0.169E-02	0.870E-04	0.935E-04
Row: 1	0.560E-05	0.380E-04	0.258E-03	0.107E-01	0.746E-01	0.157E-02	0.141E-03	0.145E-03

(c) Results obtained using $\left(\frac{du}{dn}\right)$ as a boundary condition.

Table 14: Impact of different choices of boundary condition for localized problems.

4 Features of Auditor

Many of preprocessing and postprocessing techniques outlined in Section 3 above have been assembled into a highly specialized software package called *AUDITOR*. The functionality of this package focuses on three distinct aspects of the design and analysis cycle all of which play an important part in the success of the modeling effort and the reliability of the results. Included in the design of *AUDITOR* are the following features and capabilities:

Data Structure

AUDITOR uses an object based type of data management system.

Memory Management

The data structure within *AUDITOR* uses dynamic memory allocation, thereby placing no artificial limitations on the problem size, and minimizing the storage required for smaller problems.

Two and Three-Dimensional Domains

AUDITOR can provide pre- and postprocessing of both two-dimensional and three-dimensional computational domains.

Problem Classes

AUDITOR provides pre- and processing for the following problem classes.

- Steady state heat conduction
- Linear elasticity
- Incompressible Viscous Flow
- Compressible inviscid and viscous flows

At the present time, the *AUDITOR* error estimator and solution enhancement module only accept laminar flows.

Numerical Models

The interface to *AUDITOR* has been designed to accept numerical solutions obtained from the following classes of solvers.

- Finite Element
- Finite Volume
- Finite Difference

Note, these models must be presented in a particular format to be read into the module which at the present time is based on a hexahedral cell.

Error Estimation

AUDITOR features a residual error estimator for each of the problem classes outlined above. This error estimator provides for local and global estimates of the error for either a single component of the solution or for all of the solution components collectively.

Solution Enhancement

AUDITOR contains a postprocessing option for extracting higher order continuous solution results for selected derivative components of the solution.

Mesh Preprocessing

AUDITOR contains a mesh-preprocessing module for checking the orthogonality and smoothness of a computational mesh. If problems are detected with the mesh a summary table is provided on the screen and in separate output file.

Boundary Conditions

The unique formulation of the residual error estimator within *AUDITOR* has been designed to not require that boundary conditions for the problem be supplied. This feature basically enforces the assumption the solution to the problem being solved is exact on the boundary.

Interfaces

AUDITOR provides for two formats to accept input data to be processed. These include the COMCO gridfile format and the P3/CFD restart file format. (P3/CFD is a general purpose fluid flow analysis package marketed by PDA Engineering.)

Graphical User Interface

AUDITOR was designed to operate in a workstation environment and is essentially mouse driven. The graphical interface is motif based and generally can operate in Unix environment which has X_windows or GLX libraries.

Visualization Capabilities

AUDITOR provides a wide array of graphics visualization capabilities for reviewing the solution and/or computational domain. Included in these features are

- Static and dynamic modes of operation
- Software and hardware graphics
- Visualization capabilities
  - isosurfaces
  - cutting surfaces
  - line probes
  - boundary profiles
  - velocity vectors

A summary of these features is provided in table 1.

- Numerical Models
  - 2D or 3D
  - linear or quadratic solutions for hexahedral meshes*
  - Finite Volume
  - Finite Difference
  - Finite Element
    - accepts HP meshes from P3/CFD or PHLEX
- Error Estimation Capabilities
  - Residual based error estimator for
    - Heat Conduction
    - Linear Elasticity
    - Incompressible Flow (Laminar)
    - Compressible Flow (Laminar)
  - Local cell-wise estimate of the error
  - Global estimate of the solution error
- Mesh Preconditioning
  - Identifies poorly conditioned regions of the mesh
  - Identifies tangled meshes
  - Provides a summary file of the preconditioning analysis
- Higher order solution enhancement
  - provides higher order results for various derived quantities
- Neutral File
  - Reads COMCO predefined format (see section 3)
  - Reads P3/CFD restart file
  - Reads PHLEX restart file
- Graphical User Interface
  - Completely mouse driven
  - Operates in XT mode or GLX on an SGI workstations
- Graphics/Visualization
  - Isosurfaces
  - Velocity Vectors
  - Postscript Hardcopy
  - Slices
  - XY Plotting
  - Visualization Variables include:
    - solution components
    - predefined derivatives and combinations
    - velocity magnitude
    - local error estimate
    - elements with bad jacobians

Table 15: *AUDITOR* Features and Capabilities

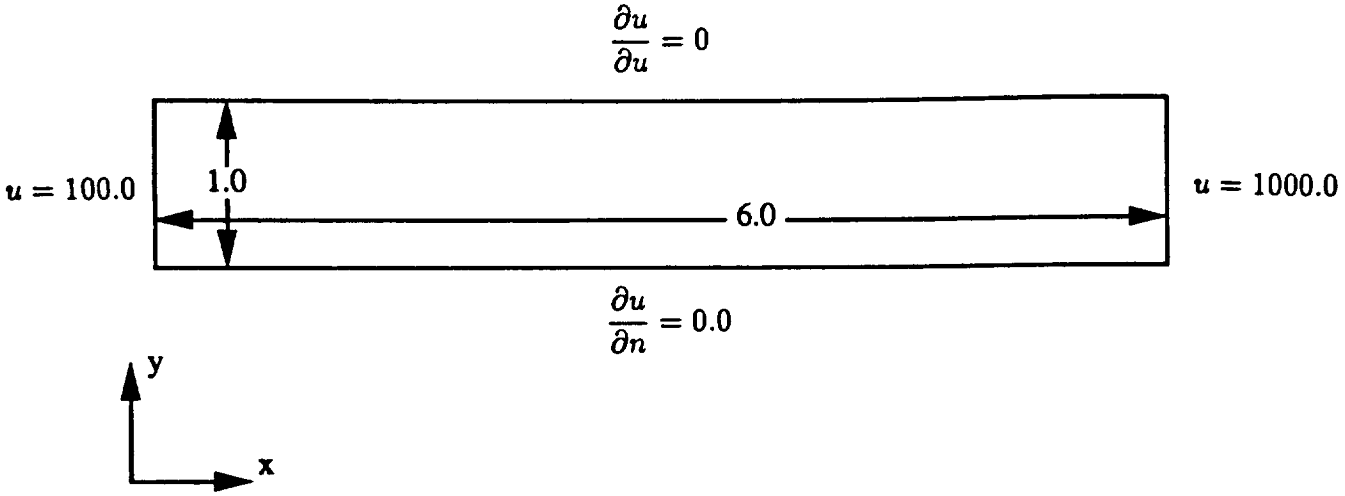

$\frac{\partial u}{\partial u} = 0$
$u = 100.0$
1.0
6.0
$u = 1000.0$
$\frac{\partial u}{\partial n} = 0.0$
y
x

Figure 5.1: Isotropic insulated bar geometry and boundary conditions for a steady state heat conduction analysis.

5 Test Problems

5.1 Introduction

This section of the final report presents selected results for a number of example problems which have been analyzed with the *AUDITOR* package. The scope of these problems range from two-dimensional steady state heat conduction to three-dimensional compressible viscous flows. The solutions to the heat conduction and linear elasticity problems are all finite element based solutions which have been obtained by the PHLEX code. This is a finite element code developed by the Computational Mechanics Company which includes hp-adaptive solution capabilities and also disposes a compatible neutral file for supplying solution data to *AUDITOR*. The solutions for the incompressible viscous flows have come from two sources, P3/CFD, a general purpose finite element CFD code marketed by PDA Engineering, and from the TEACH code developed at the Imperial College of Science and Technology in London. Finally, the solutions for the compressible flows have been taken from P3/CFD, and from a finite volume code, JCODE, which is based on Jameson's finite volume Runge-Kutta time stepping scheme.

On a final note before beginning a discussion of the test problems; the graphics and visualization capabilities in the *AUDITOR* package are based on a three-dimensional model and thus even for two-dimensional solutions the graphics will appear as extruded three-dimensional plots with a unit thickness.

5.2 Steady State Heat Conduction

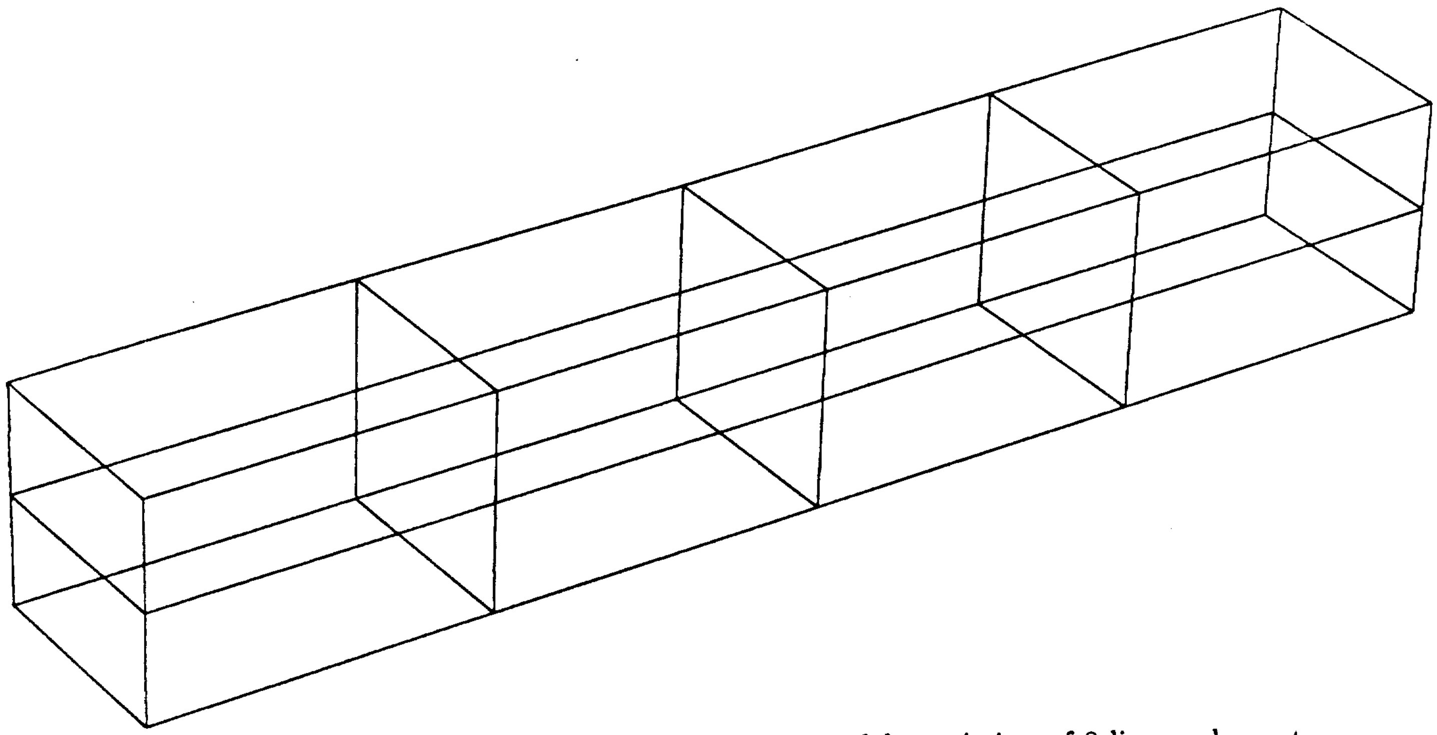


Figure 5.2: Isotropic insulated bar, finite element model consisting of 8 linear elements.

5.2.1 Isotropic Insulated Bar (PHLEX - finite element solution)

The first sample problem presented herein is a steady heat conduction analysis of two-dimensional rectangular bar. The geometry and boundary conditions used to solve this problem with the PHLEX code are shown in Figure 5.1. Summarizing the boundary condition data shown in this figure, there is a prescribed temperature of 100° on the left face, a prescribed temperature of 1000° on the right face, and insulated boundaries (zero heat flux) on the upper and lower faces. The thermal conductivity is assumed to be independent of the coordinate location and is arbitrarily selected to be 1.0. For this problem, an exact solution is known and is given by the following expression

$$u(x, y, z) = 100 + 900 * x/l$$

Here x is the axial coordinate length measured from the left end of the bar and l is the length of the bar.

Numerical Model

The geometry, shown in Figure 5.1, has initially been discretized using 8 linear finite elements with 4 elements in the axial direction and two elements through the thickness, see Figure 5.2.

AUDITOR Analysis

This problem was solved using the PHLEX code which generates a neutral file format which can be directly read into *AUDITOR*. As this mesh is quite simple, there is basically no need to preprocess the mesh so this phase of the analysis will be omitted. The features of interest for this linear steady state problem are the temperature distribution and the global and local error distribution. To ensure that the numerical has been correctly read into the database, an isosurface plot of the temperature distribution is shown in Figure 5.3. (Again, note that the graphics package is a three-dimensional package which shows the computational domain with the geometry and solutions expand with a unit value in the third dimension.) From this figure we observe that the temperature variation in the mesh has exactly captured the linear variation of the temperature from 100 to 1000. Activating the residual error estimator, the estimated error in the solution is zero verifying the residual error estimator in operating correctly.

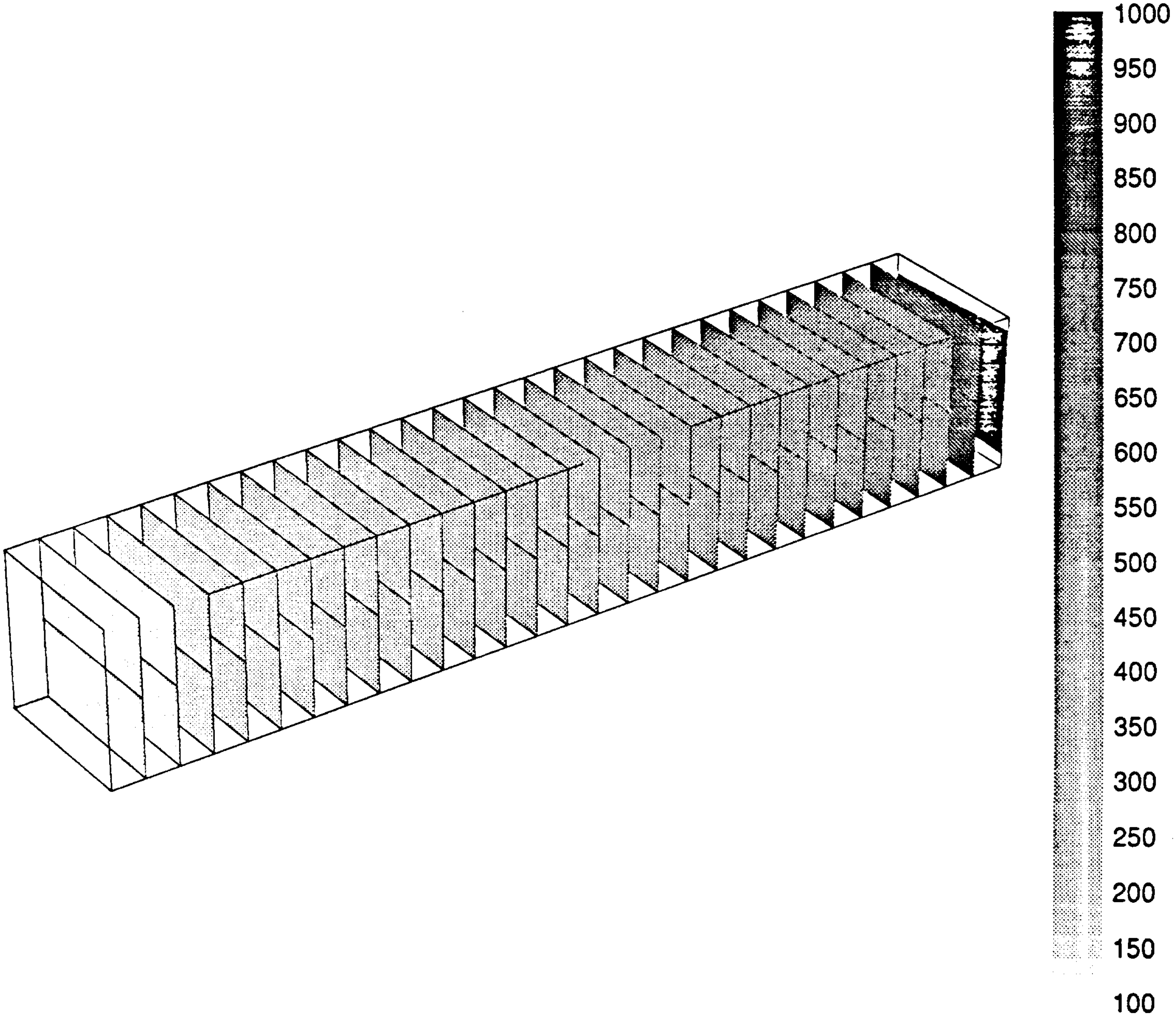


Figure 5.3: Temperature isosurfaces for the insulated bar.

5.2.2 Isotropic Bar With Prescribe Source (PHLEX - finite element solution)

A second example based on a simple rectangular geometry, shown in Figure 5.4, has been used in the solution of a three-dimensional heat conduction problem with a slightly different set of boundary conditions and a heat source term. Summarizing the boundary condition data shown in this figure, all of the boundaries of the three-dimensional bar have a prescribed temperature of zero and the heat source term has been selected to have the following quadratic volumetric distribution

$$
\begin{aligned}
s(x,y,z) = \; & -310 * ((-2) * \quad (1-y) * y(3-z) * z \\
& +(2-x) * x * (-2) * \quad (3-z) * z \\
& +(2-x) * x * (1-y) * y * (-2)) \\
& -170 * (\, (4-6*x) * (1-y) * y * (3-z) * z \\
& +(2-x) * x * *2 * \; (-2) * \quad (3-z) * z \\
& +(2-x) * x * *2 * (1-y) * y * (-2)) \\
& -290 * (\, (-2) * \quad (1-y) * y * *2 * (3-z) * z \\
& +(2-x) * x * (2-6*y) * (3-z) * z \\
& +(2-x) * x * (1-y) * y * *2 * (-2)) \\
& -130 * (\, (-2) * \quad (1-y) * y * (3-z) * z * *2 \\
& +(2-x) * x * (-2) * \quad (3-z) * z * *2 \\
& +(2-x) * x * (1-y) * y * (6-6*z)) \\
& -230 * (\, (-2) * (1-y) * y * *2 * (3-z) * z * *2 \\
& +(2-x) * x * (2-6*y)*) * (3-z) * z * *2 \\
& +(2-x) * x * (1-y) * y * *2 * (6-6*z)) \\
& -110 * (\, (4-6*x) * (1-y) * y * (3-z) * z * *2 \\
& +(2-x) * x * *2 * \; (-2) * \quad (3-z) * z * *2 \\
& +(2-x) * x * *2 * (1-y) * y * (6-6*z)) \\
& -190 * (\, (4-6*x) * (1-y) * y * *2 * (3-z) * z \\
& +(2-x) * x * *2 * \; (2-6*y) * \quad (3-z) * z \\
& +(2-x) * x * *2 * (1-y) * y * *2 * (-2)) \\
& -100 * (\, (4-6*x) * (1-y) * y * *2 * (3-z) * z * *2 \\
& +(2-x) * x * *2 * \; (2-6*y) * (3-z) * z * *2 \\
& +(2-x) * x * *2 * (1-y) * y * *2 * (6-6*z))
\end{aligned}
$$

The selection of this source distribution allows an exact solution to be determined which is given by the following expression:

$$u(x,y,z) = 0 \qquad \text{in } \partial\Omega$$

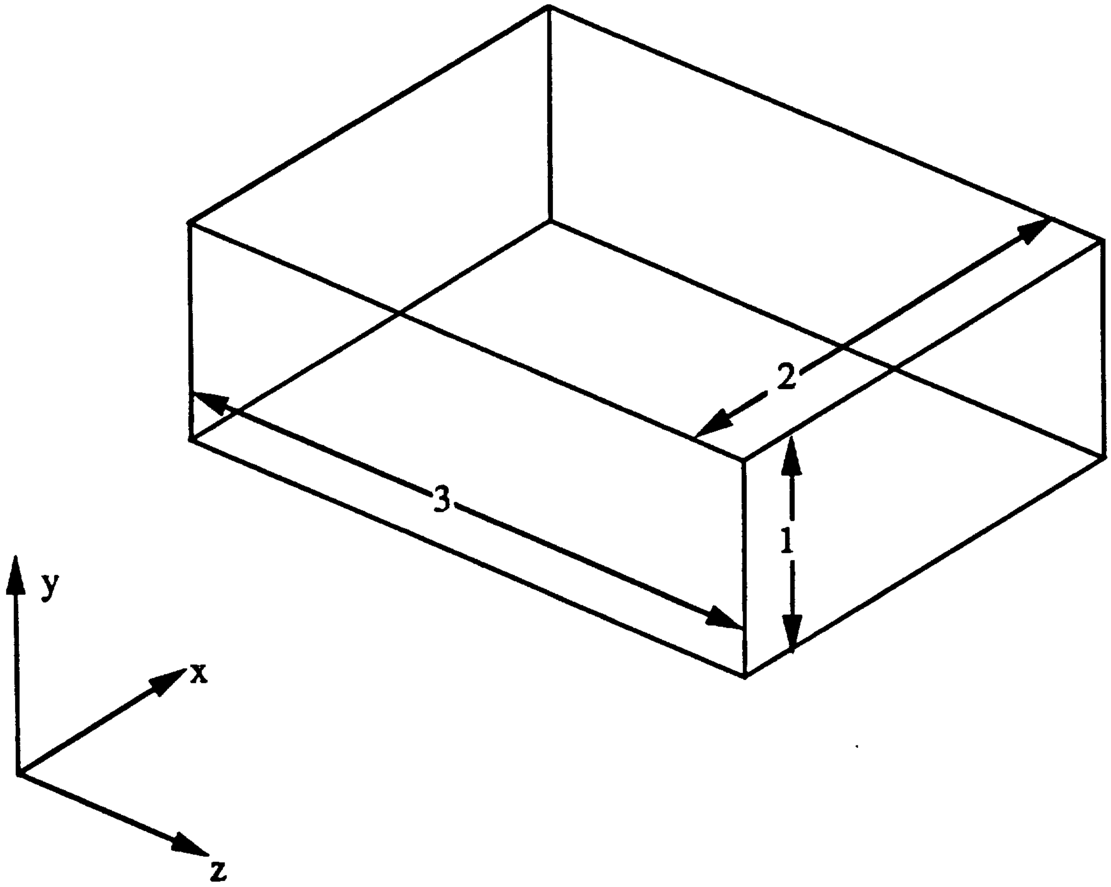


Figure 5.4: Isotropic bar with prescribed volumetric source, geometry and boundary conditions.

$$
\begin{aligned}
u(x,y,z) = \; & -310*(\,(2-x)*x \;*\; (1-y)*y \;*(3-z)*z) \\
& -170*(\,(2-x)*x \;*\; (1-y)*y \;*(3-z)*z)*x \\
& -290*(\,(2-x)*x \;*\; (1-y)*y \;*(3-z)*z)*y \\
& -130*(\,(2-x)*x \;*\; (1-y)*y \;*(3-z)*z)*z \\
& -230*(\,(2-x)*x \;*\; (1-y)*y \;*(3-z)*z)*yz \\
& -110*(\,(2-x)*x \;*\; (1-y)*y \;*(3-z)*z)*xz \\
& -190*(\,(2-x)*x \;*\; (1-y)*y \;*(3-z)*z)*xy \\
& -100*(\,(2-x)*x \;*\; (1-y)*y \;*(3-z)*z)*xyz
\end{aligned}
$$

The thermal conductivity is again assumed to be independent of the coordinate locations and we have selected it to be 1.0. Note, to run this problem using *AUDITOR*, an environment variable which triggers this body force needs to be set as follows:

```
setenv POISSON_TEST ON
```

Numerical Model

The geometry, shown in Figure 5.4, has initially been discretized using 16 finite elements with 4 elements in the axial (x) direction and 4 elements in the cross section. The PHLEX analysis for this problem was carried out in three stages which consisted of calculating the solution on a linear finite element mesh, an isotropic quadratic mesh, and an isotropic cubic mesh. The goal with the *AUDITOR* analysis is to observe the change in the residual error as the spectral order of the mesh is increased. Note, that since a very specific volumetric source term was selected, the exact solution is a cubic polynomial which implies that the *AUDITOR* code should predict a zero error for the third order PHLEX analysis.

AUDITOR Analysis

The meshes used in all three of the solution sequences are again quite simple and no preprocessing of the mesh is required. The features of interest for this steady state analysis include the temperature distribution within the domain, and a monitoring of the global error as the spectral order of the mesh is increased. To ensure that the numerical solutions have been correctly read into the database, a plot of the isosurfaces of the temperature of each of the models is shown in Figure 5.5 - 5.7. Calculating the residual errors for each of the meshes, the error in the linear solution is 261%, the error in the quadratic mesh is 8.3 %, and the error in the cubic solution is zero.

5.2.3 Connecting Rod (PHLEX - finite element solution)

The final steady heat conduction analysis presented here is of a three-dimensional connecting rod. The geometry and boundary conditions used in the PHLEX simulation are shown in

Figure 5.8. Summarizing the boundary conditions, there is a prescribed temperature of 500° on the piston pin contact surface, left most circular arc in Figure 5.8, a prescribed temperature of 250° on the crank shaft contact surface, right most quarter circular arc, and zero normal heat flux on the remaining boundaries. The volumetric source term is set to zero and the thermal conductivity is specified to be a constant of 1.0.

Numerical Model

The geometry, shown in Figure 5.8, has initially been discretized using 194 linear finite elements which have been enriched to second order for the solution phase, see Figure 5.9.

AUDITOR Analysis

The first phase of the *AUDITOR* analysis for this test problem accesses the *AUDITOR* pre-processor to gather diagnostics concerning the mesh quality. Figure 5.10 shows a copy of the diagnostics which appear on the screen as a result of this preprocessing phase. Of note here are the following: there are no elements with bad surface or volume jacobians, the critical angle selected for orthonogality testing was 30° and 8 cells in the mesh were found to have angles which are outside the critical boundary of 30° and 150°, the largest and smallest included angles in the mesh are 177.19 and 15.87 respectively, and the largest volumetric and surface jacobian ration was found to be 0.069503. This final value, the jacobian ration, in general, gives an indication that at least one of the elements used in the computations is severely distorted locally and could possibly cause numerical problems.

In the second phase of the analysis, the global and local error distributions were calculated. Based on the quadratic finite element approximation to the solution, the residual error estimator indicates that there is a 1.56% global error in the solution with a maximum local error of 0.52%. Figure 5.11 shows the corresponding distribution of the residual error projected onto the surface of the connecting rod. Interpreting the color coding in this figure, it appears that the highest errors in the solution occur near the intersections of the connecting rod and the Piston contact surfaces. A zoom of this region is shown in Figure 5.12. For reference, we have also included a copy of the temperature distribution, Figure 5.13 which shows the maximum temperature of 500° on the piston contact surface, and a temperature of 250° on the crank shaft contact surface.

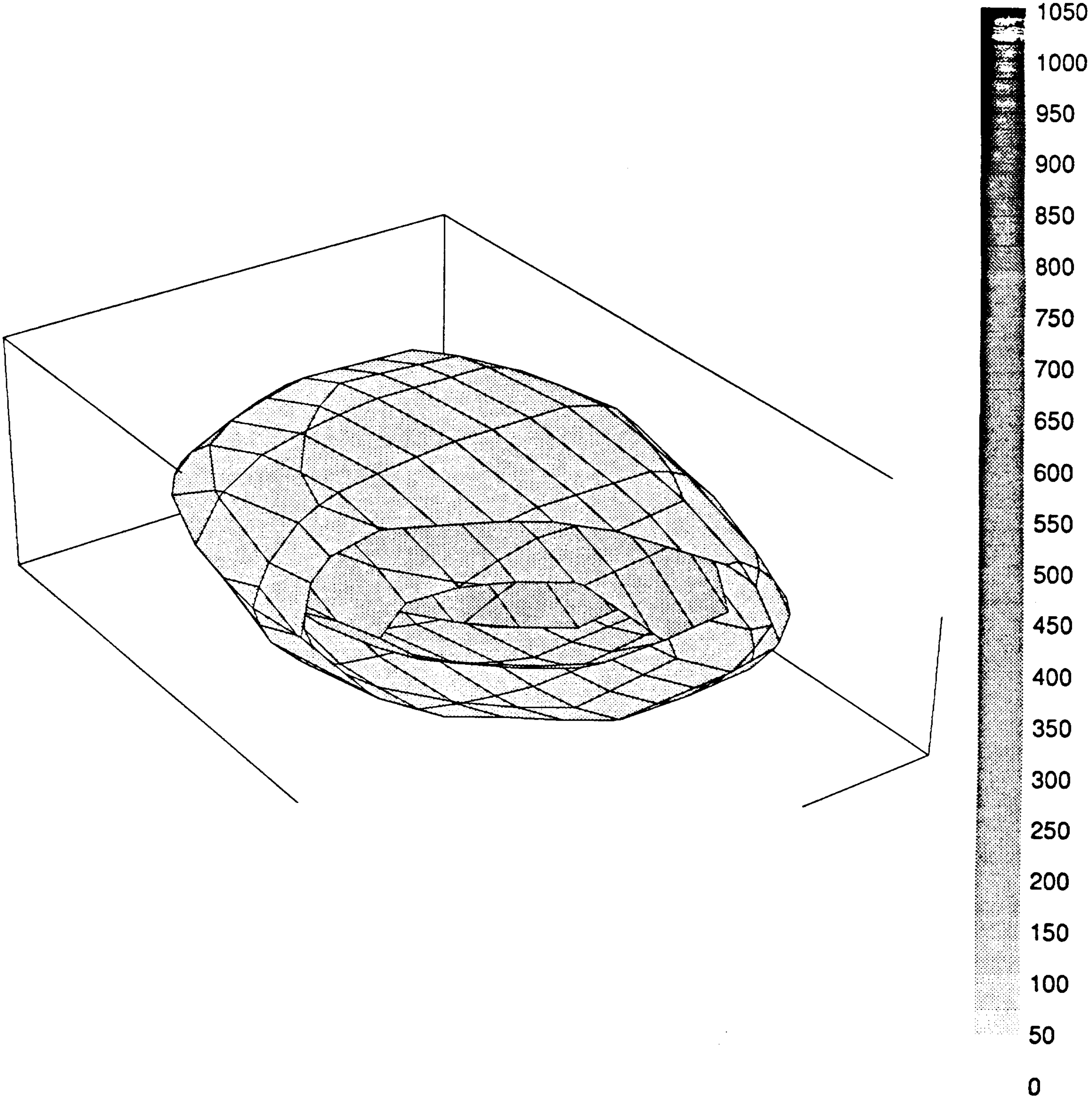


Figure 5.5: Temperature isosurface for an isotropic bar with a prescribed source for a linear finite element solution

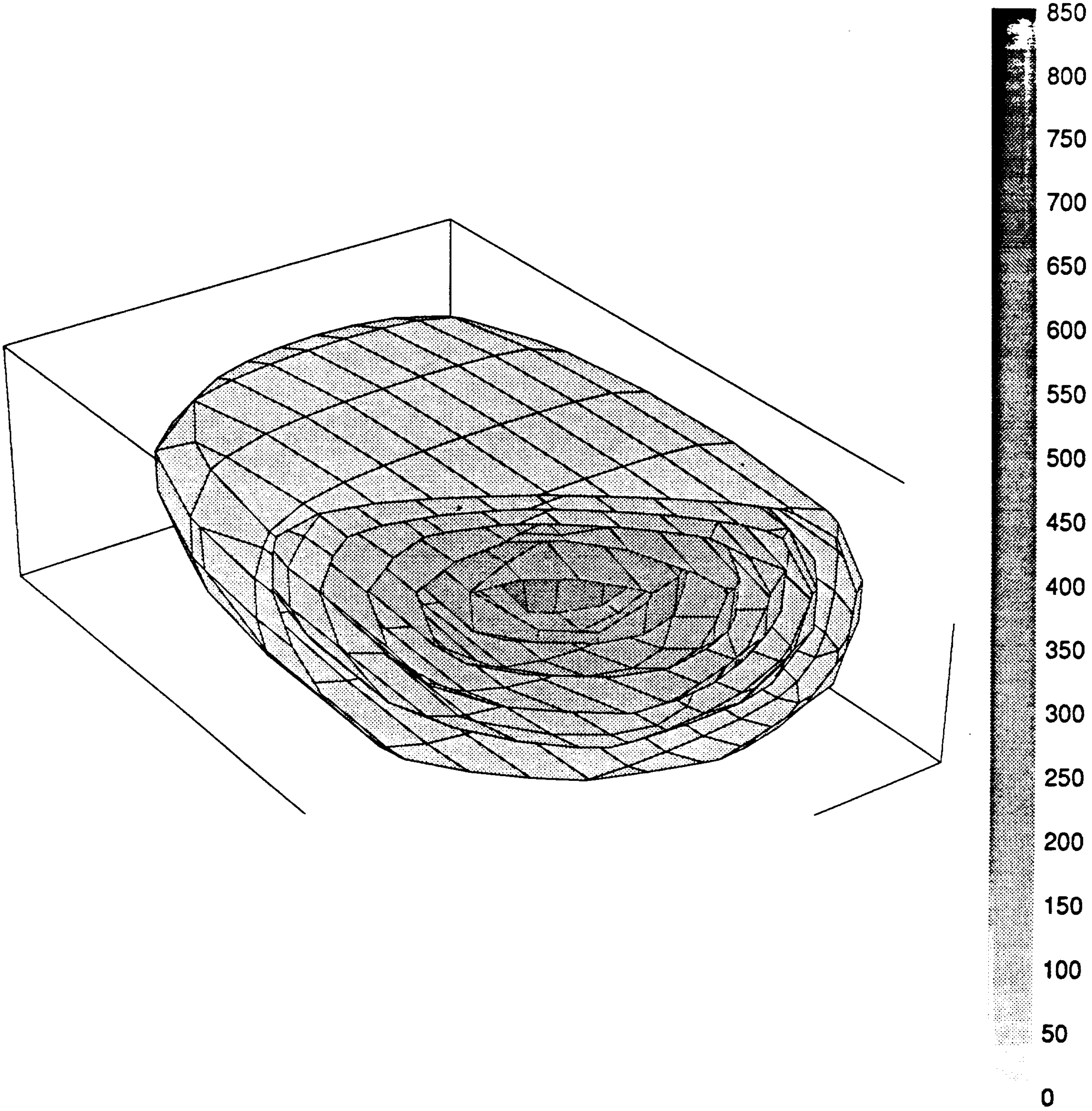


Figure 5.6: Temperature isosurface for an isotropic bar with a prescribed source for a quadratic finite element solution

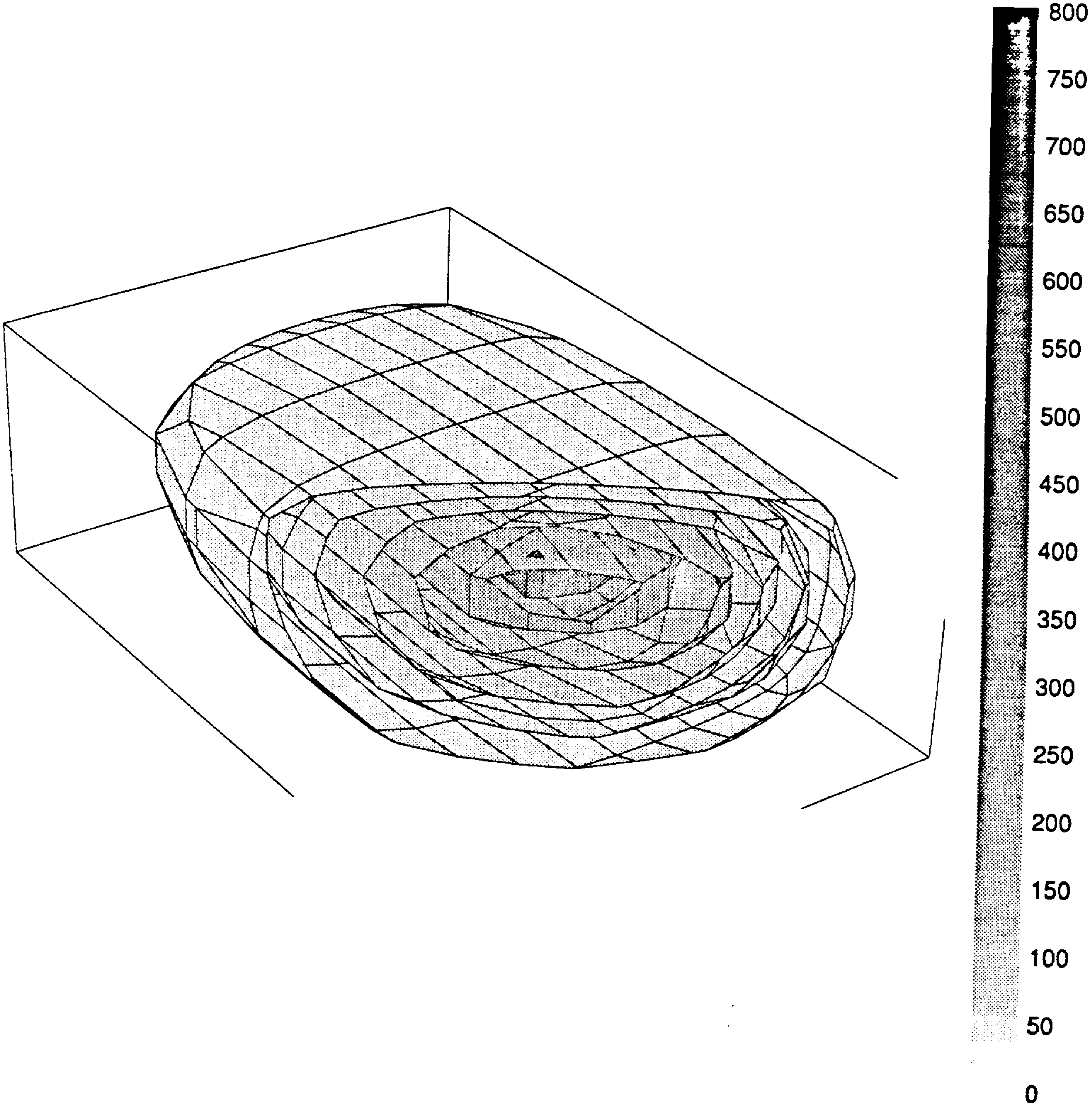


Figure 5.7: Temperature isosurface for an isotropic bar with a prescribed source for cubic finite element solution

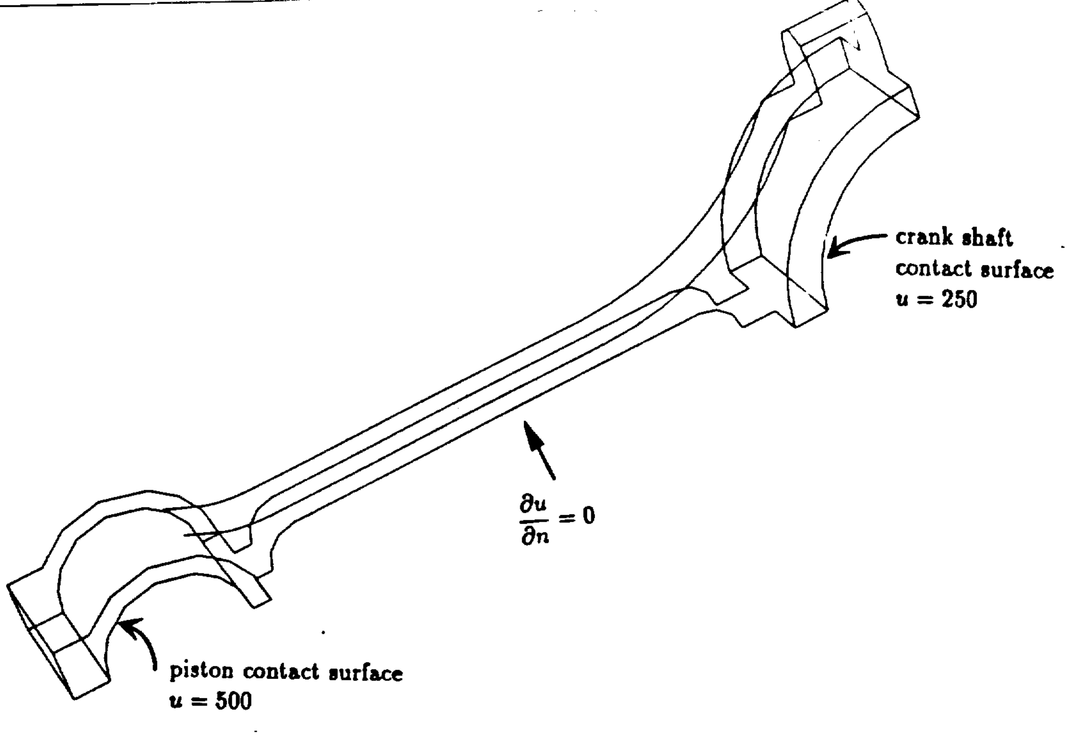


Figure 5.8: Three-dimensional connecting rod, geometry and boundary conditions

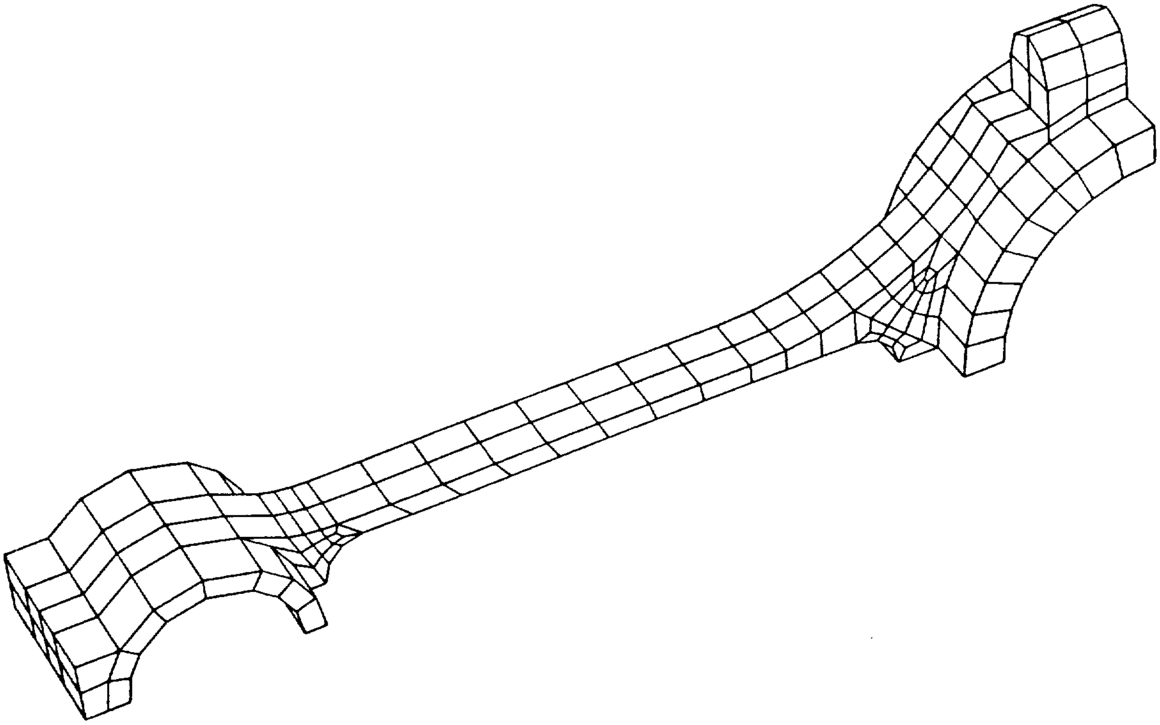
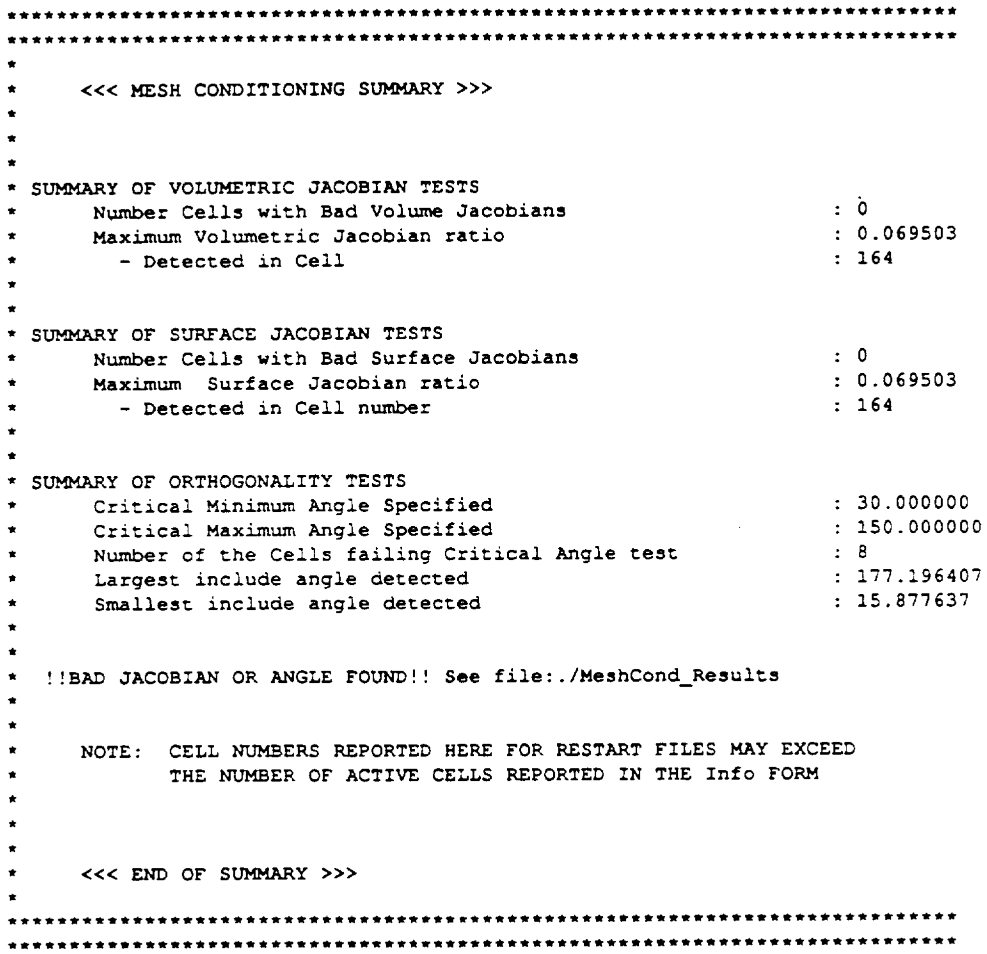


Figure 5.9: Finite element for a connecting rod mesh consisting of 194 elements and 2417 degrees of freedom



```
****************************************************************************
****************************************************************************
*
*     <<< MESH CONDITIONING SUMMARY >>>
*
*
*
* SUMMARY OF VOLUMETRIC JACOBIAN TESTS
*      Number Cells with Bad Volume Jacobians                     : 0
*      Maximum Volumetric Jacobian ratio                          : 0.069503
*        - Detected in Cell                                       : 164
*
*
* SUMMARY OF SURFACE JACOBIAN TESTS
*      Number Cells with Bad Surface Jacobians                    : 0
*      Maximum  Surface Jacobian ratio                            : 0.069503
*        - Detected in Cell number                                : 164
*
*
* SUMMARY OF ORTHOGONALITY TESTS
*      Critical Minimum Angle Specified                           : 30.000000
*      Critical Maximum Angle Specified                           : 150.000000
*      Number of the Cells failing Critical Angle test            : 8
*      Largest include angle detected                             : 177.196407
*      Smallest include angle detected                            : 15.877637
*
*
*  !!BAD JACOBIAN OR ANGLE FOUND!! See file:./MeshCond_Results
*
*
*     NOTE:  CELL NUMBERS REPORTED HERE FOR RESTART FILES MAY EXCEED
*            THE NUMBER OF ACTIVE CELLS REPORTED IN THE Info FORM
*
*
*
*     <<< END OF SUMMARY >>>
*
****************************************************************************
****************************************************************************
```

Figure 5.10: Preprocessor analysis of the computational mesh for a connecting rod

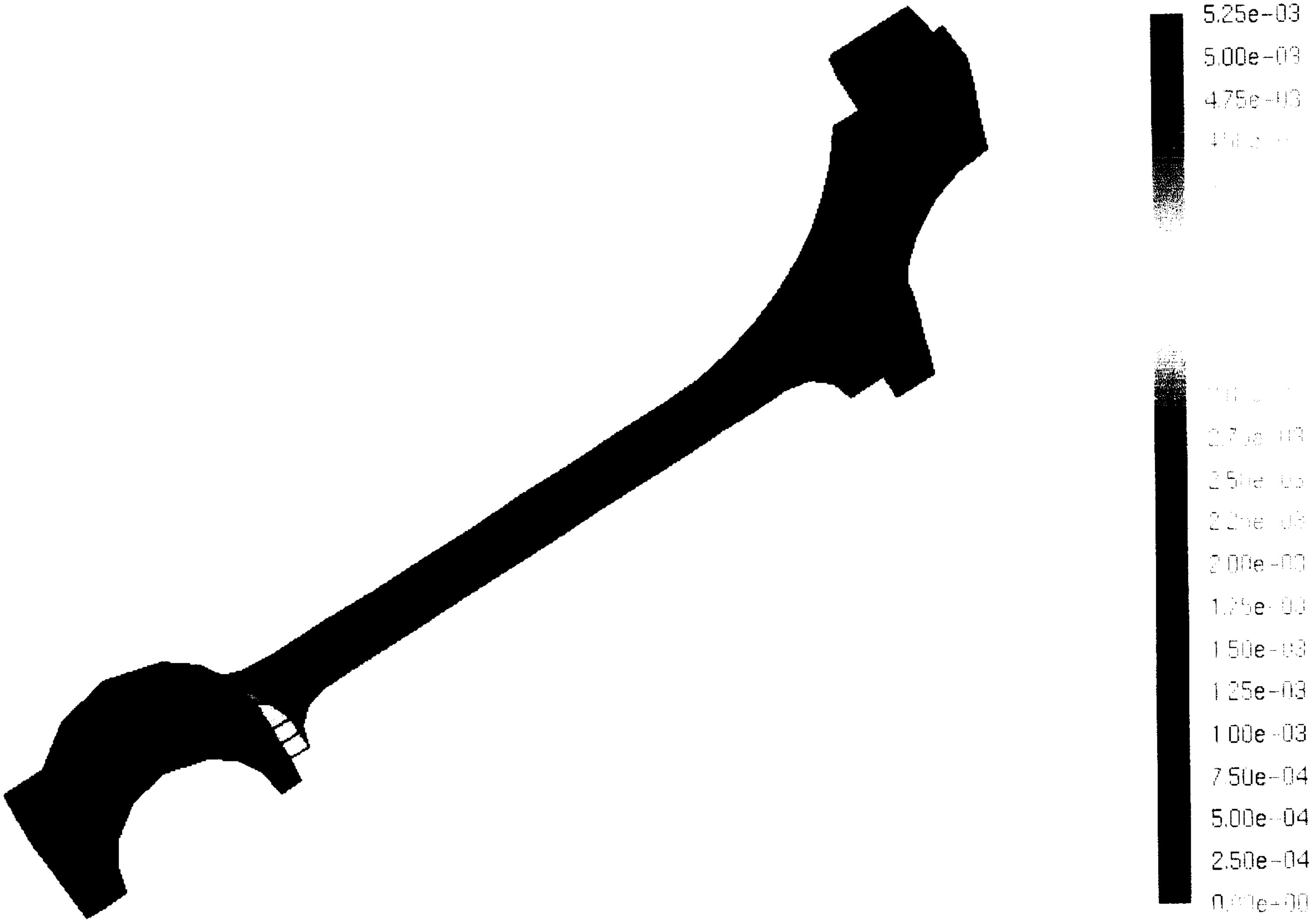


Figure 5.11: Residual error estimation distribution for a steady state solution of a three-dimensional connecting rod

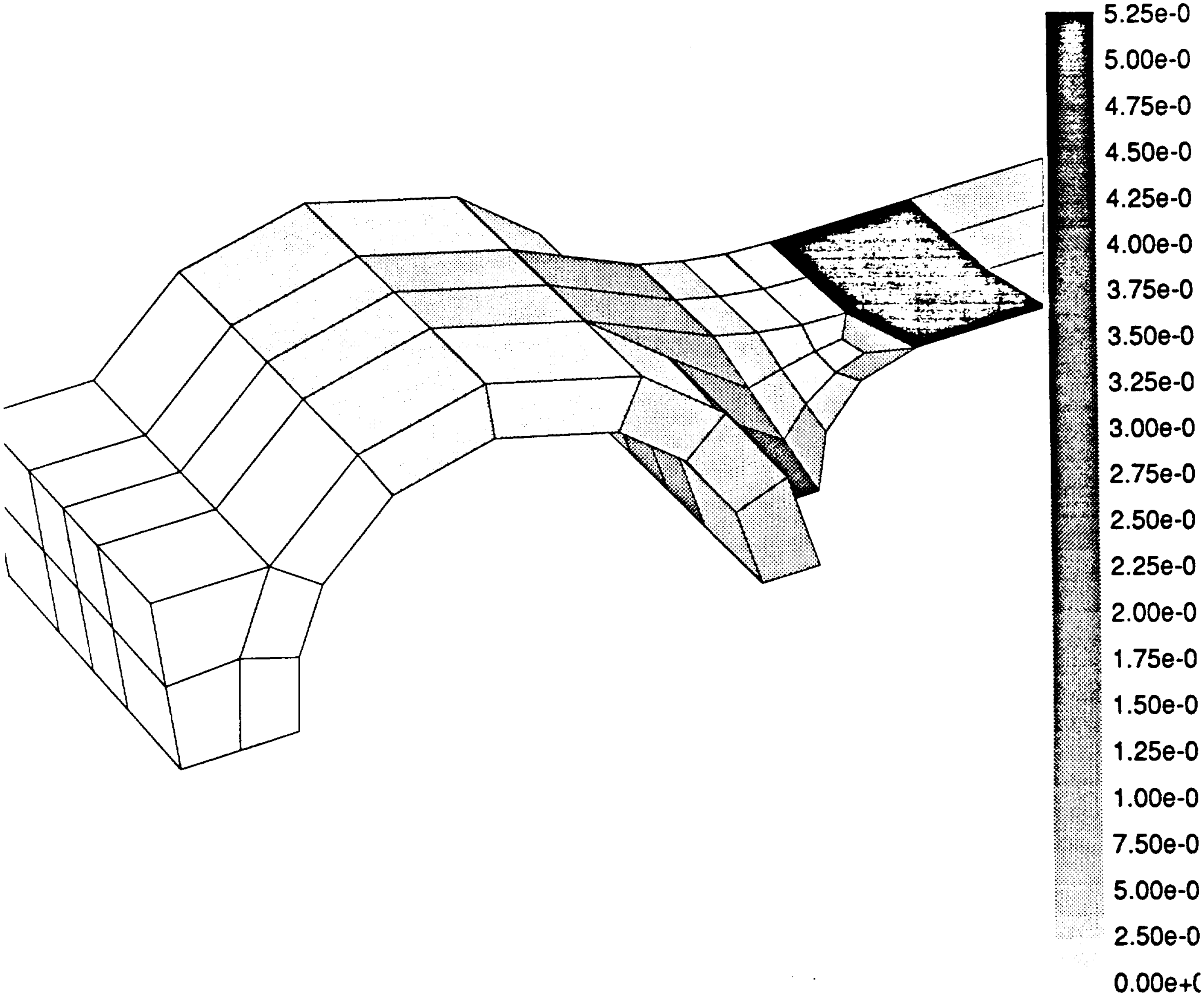


Figure 5.12: Zoom of the error estimation distribution in the piston contact region

5.3 Linear Elasticity

5.3.1 Isotropic Bar (PHLEX - finite element solution)

The first linear elasticity problem presented herein is a three-dimensional rectangular bar in simple tension. The geometry and boundary conditions used to solve this problem with the PHLEX code are shown in Figure 5.14. Summarizing the boundary conditions, there are no penetration boundaries along the left, bottom, and back surfaces, a uniform surface traction 1000 lbs. in the horizontal x direction on the right face and zero surface tractions on the remaining surfaces. The body force terms are zero and the following material constants apply;

$$\begin{array}{rcrcl} \lambda = & \text{Lamae's constant} & = & \text{5,617,280 psi} \\ \text{G=} & \text{Shear modulus} & = & \text{2,407,407 psi} \end{array}$$

The exact solution is easily calculated to be a linear function of the coordinate locations with the following components;

$$\begin{aligned} u(x,y,z) &= 9.230770e^{-4} * x/l \text{ (in.)} \\ v(x,y,z) &= - 5384516e^{-5} * y/l \text{ (in.)} \\ w(x,y,z) &= - 5.384516e^{-5} * z/l \text{ (in.)} \end{aligned}$$

Here, x is the axial coordinate location along the bar, y and z are the transverse coordinate directions, and l is the length of the length of the bar.

Numerical Model

The geometry for this example, shown in Figure 5.14, has initially been discretized using 16 linear finite elements with 4 elements in the axial direction and four elements in the cross section, see Figure 5.15.

AUDITOR Analysis

Due to the simplicity of the computational mesh, and solution, the only phases in the *AUDITOR* analysis that are of interest are the error estimation phase and possibly the solution enhancement phase. Before activating either of these options, a plot of the isosurfaces for the u-displacement component have been extracted as shown in Figure 5.16. This figure shows that the axial displacements vary linearly from 0.0 inches on the left end to 9.23077 e^{-4} (in.) at the right end, which corresponds to the exact solution given above. Activating the residual error estimator we find that the estimated error in the solution is zero verifying that residual error estimator is operating correctly. For this particular case, the stress distribution is constant, $\sigma_{xx} = 1000$ psi (all other $\sigma_{ij} = 0.0$) and thus initiation of the postprocessing solution enhancement shows no improvement in any of the stress components.

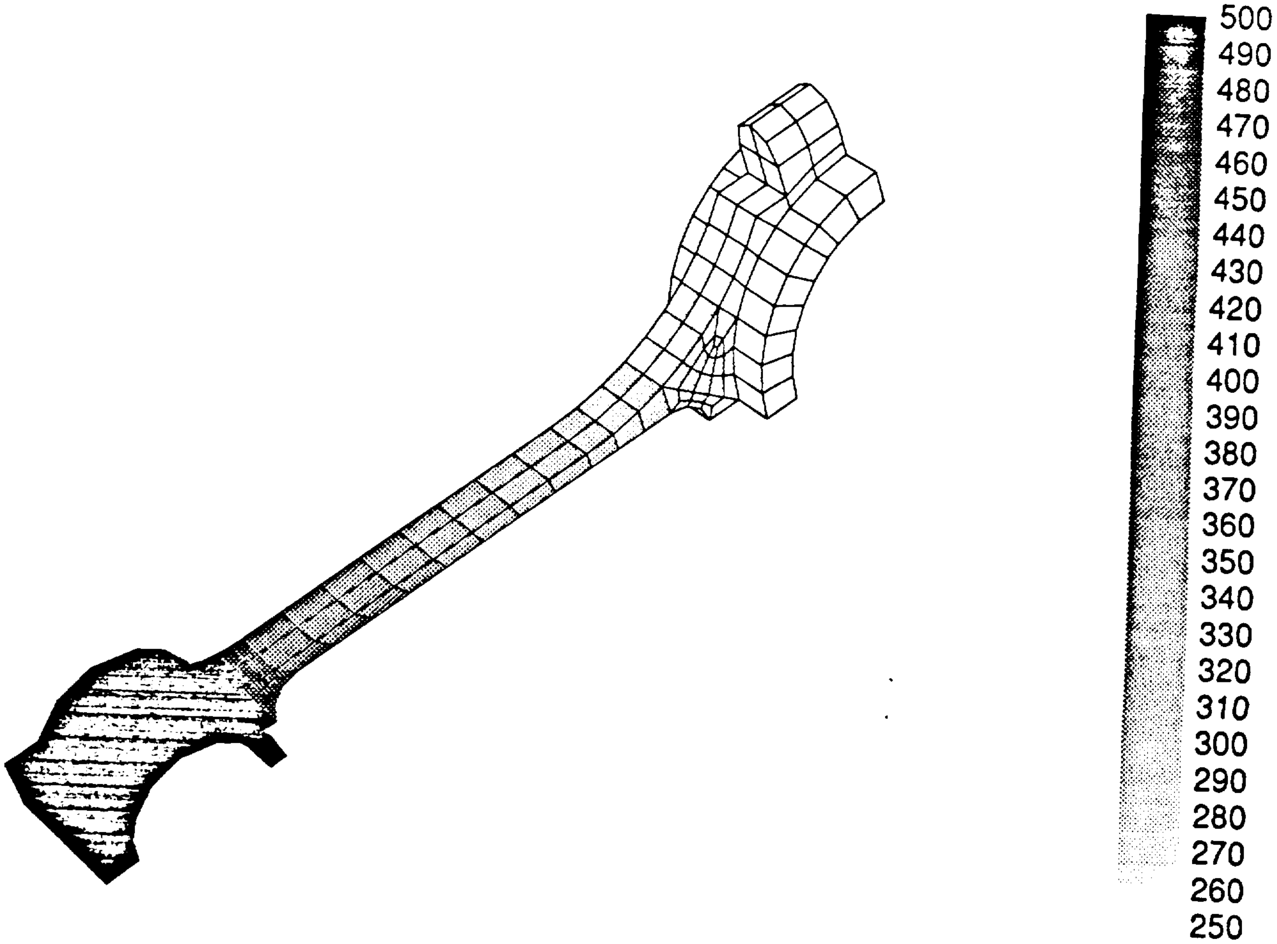


Figure 5.13: Surface temperature distribution for an isotropic connecting rod

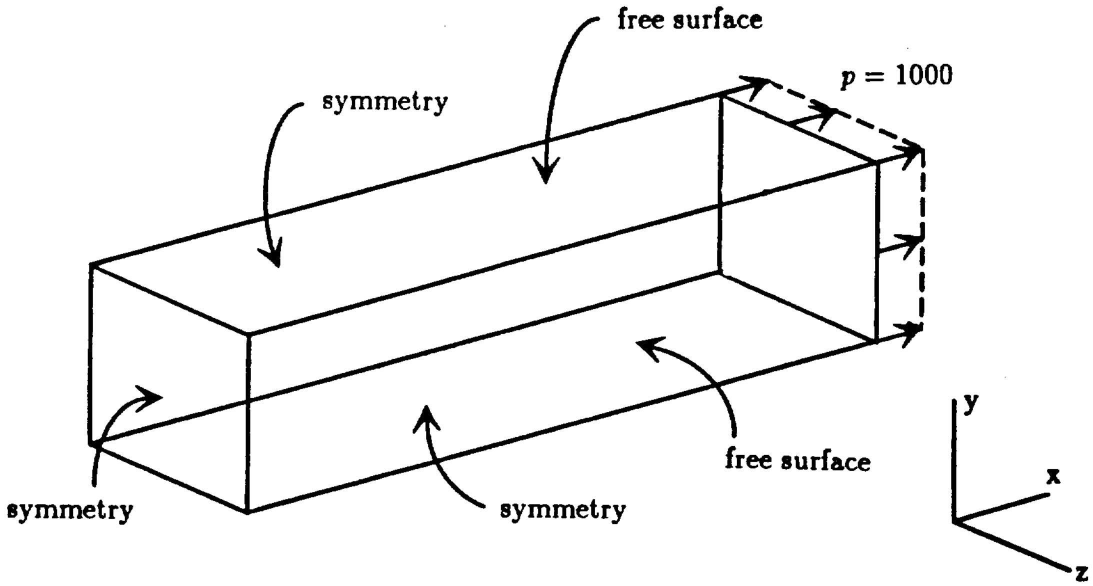


Figure 5.14: Geometry and boundary conditions for isotropic bar under simple tension

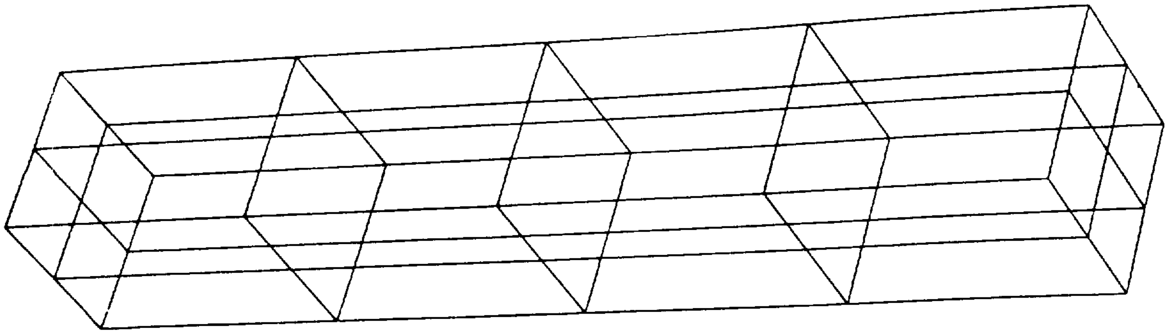


Figure 5.15: Finite element mesh for a linear elastic bar under simple tension

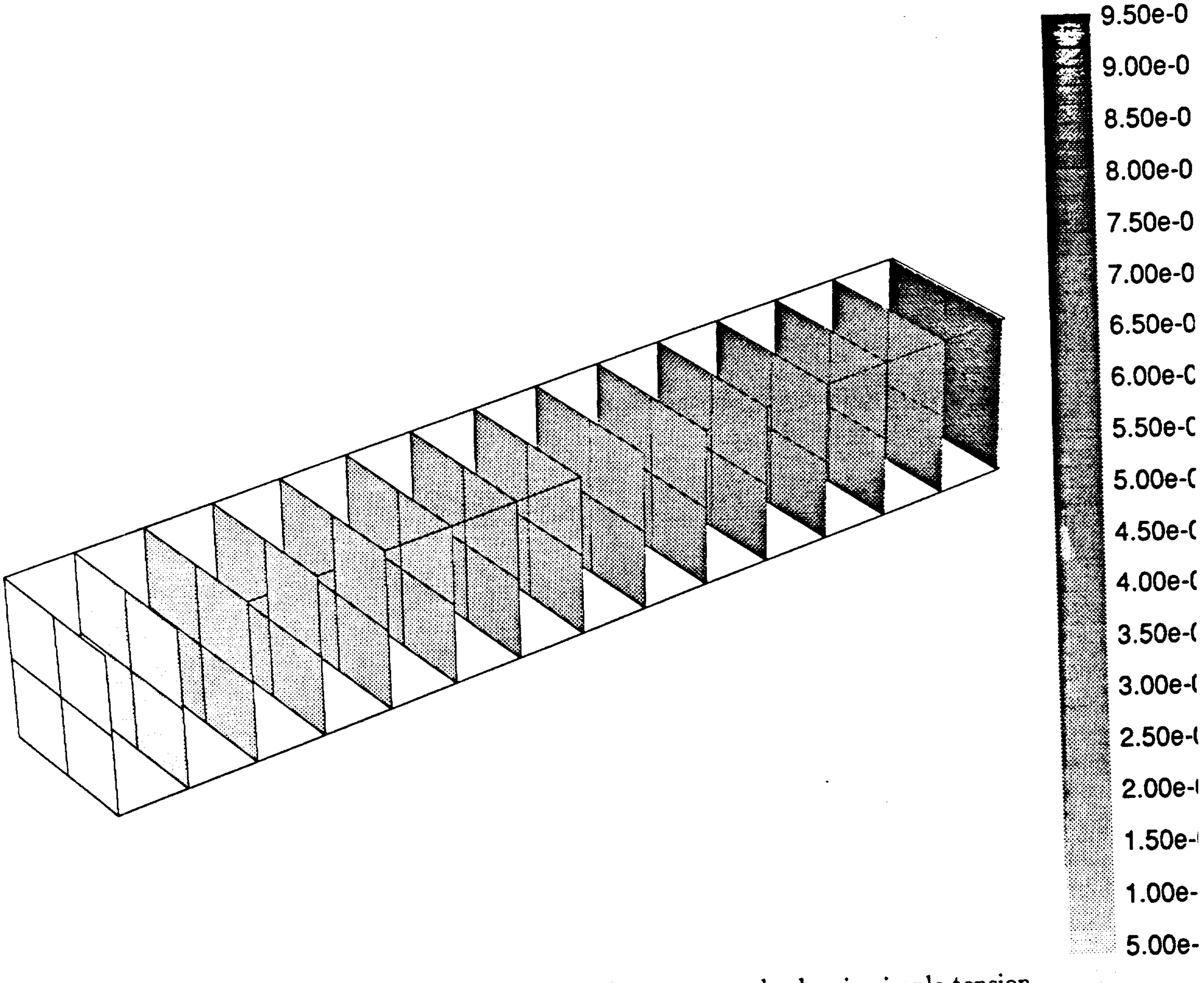


Figure 5.16: U-displacement isosurfaces for a rectangular bar in simple tension

5.3.2 Wheel Rim (PHLEX - finite element solution)

The final linear elasticity problem presented herein is for a three-dimensional analysis of wheel rim on a Pontiac Fiero. The geometry and boundary conditions used in the PHLEX simulation are shown in Figures 5.17 and 5.18. Summarizing the boundary conditions in these figures, there is a prescribed lateral bead pressure of 313 psi due to tire pressurization, a vertical load of 123.4 psi on the rim due to weight of the automobile, fixed displacements at the lug-nut locations, and free surfaces elsewhere. The body force terms are prescribed as zero and we have selected the following material constants:

$$\begin{aligned} \lambda = &\quad \text{Lamae's constant} &= \quad 5{,}617{,}280 \text{ psi} \\ \text{G}= &\quad \text{Shear modulus} &= \quad 2{,}407{,}407 \text{ psi} \end{aligned}$$

Numerical Model

The geometry, shown in Figures 5.17 and 5.18, has initially been discretized using 570 linear finite elements with 1300 degrees of freedom, see Figure 5.19. Note, that the model shown in Figure 5.19 contains symmetry only about the $z = 0$ and $y = 0$ planes.

AUDITOR Analysis

The first phase of the *AUDITOR* analysis for this test problem accesses the preprocessor to gather diagnostics concerning the mesh quality. Figure 5.20 shows a copy of the diagnostics which appear on the screen as a results of this preprocessing phase. Of note here are the following: there are no elements with bad surface or volume jacobians, the largest and smallest included angles in the mesh are 136.0° and 25.5° respectively, and the largest jacobian ratio was found to be 0.228 which indicates only moderate distortions of mesh and which should not effect the computations.

In the second phase of the analysis we have calculated the global and local error distributions. For this linear finite element solution the residual error estimator indicates that there is an 532% global error in the solution with a maximum local error of 75.5%. Figure 5.21 shows the corresponding distribution of the residual error projected onto the wheel surface. Based on the color coding of this figure, the highest errors in the solution are occurring almost directly below the wheel axle at the intersection of the cylinder drum and the tie bar. Note, that the error distribution is symmetric about the $z = 0$ plane but exhibits no other symmetry due to the loading pattern and orientation of the tie bars.

To complete the analysis of the wheel rim, we have accessed the postprocessing options to extract higher order results for the Von Mises stress. As a reference point we have plotted isosurfaces on the boundary of the current distribution of the Von Mises stress as shown in Figure 5.22. Note, from this figure an the diagnostics in the isosurface plotting window, the stresses are discontinuous across element interfaces, and the maximum stress value is 2520 psi. Figure 5.23 shows the corresponding Von Mises stress distribution after the postprocessor has been called. Based on the enhancement procedure, the Von Mises stresses are now continuous and the maximum stress value has been slightly relieved.

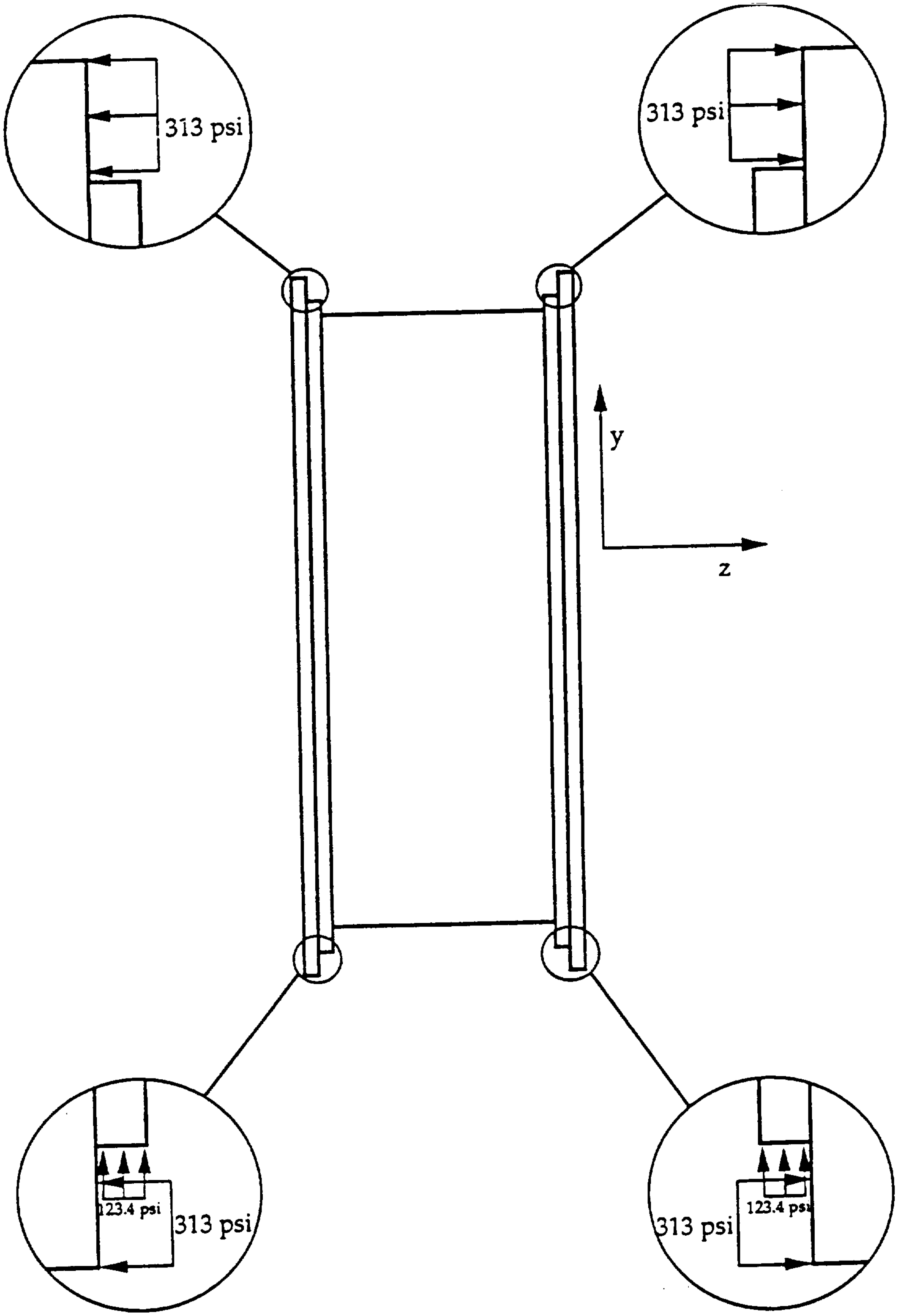


Figure 5.17: Geometry and loading configuration for an automobile wheel subjected to static tire pressure and vertical bead loading. Side view showing tire pressure loads (313 psi loads) and vertical bead loads (123.4 psi loads.)

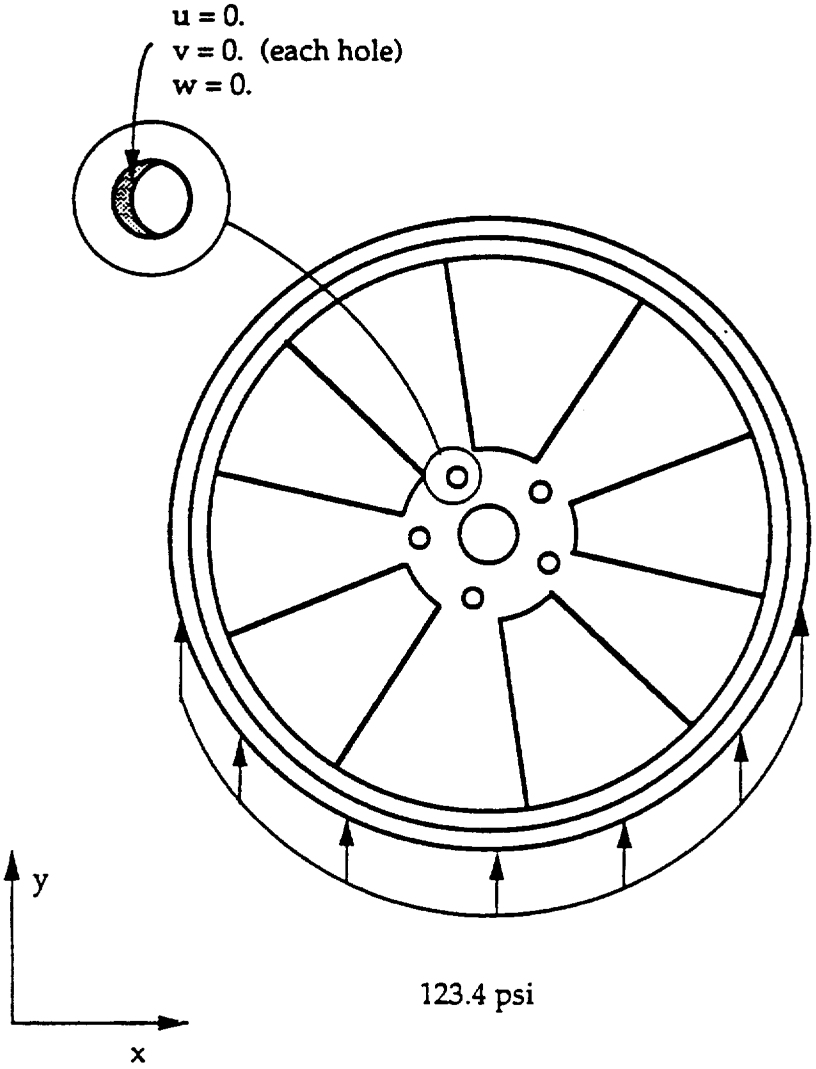


Figure 5.18: Geometry and loading configuration an automobile wheel subjected to static tire pressure and vertical bead loading. Front view showing radial the distribution of the vertical bead load. Tire pressure loads are not shown as they are uniformly distributed around the rim of the wheel.

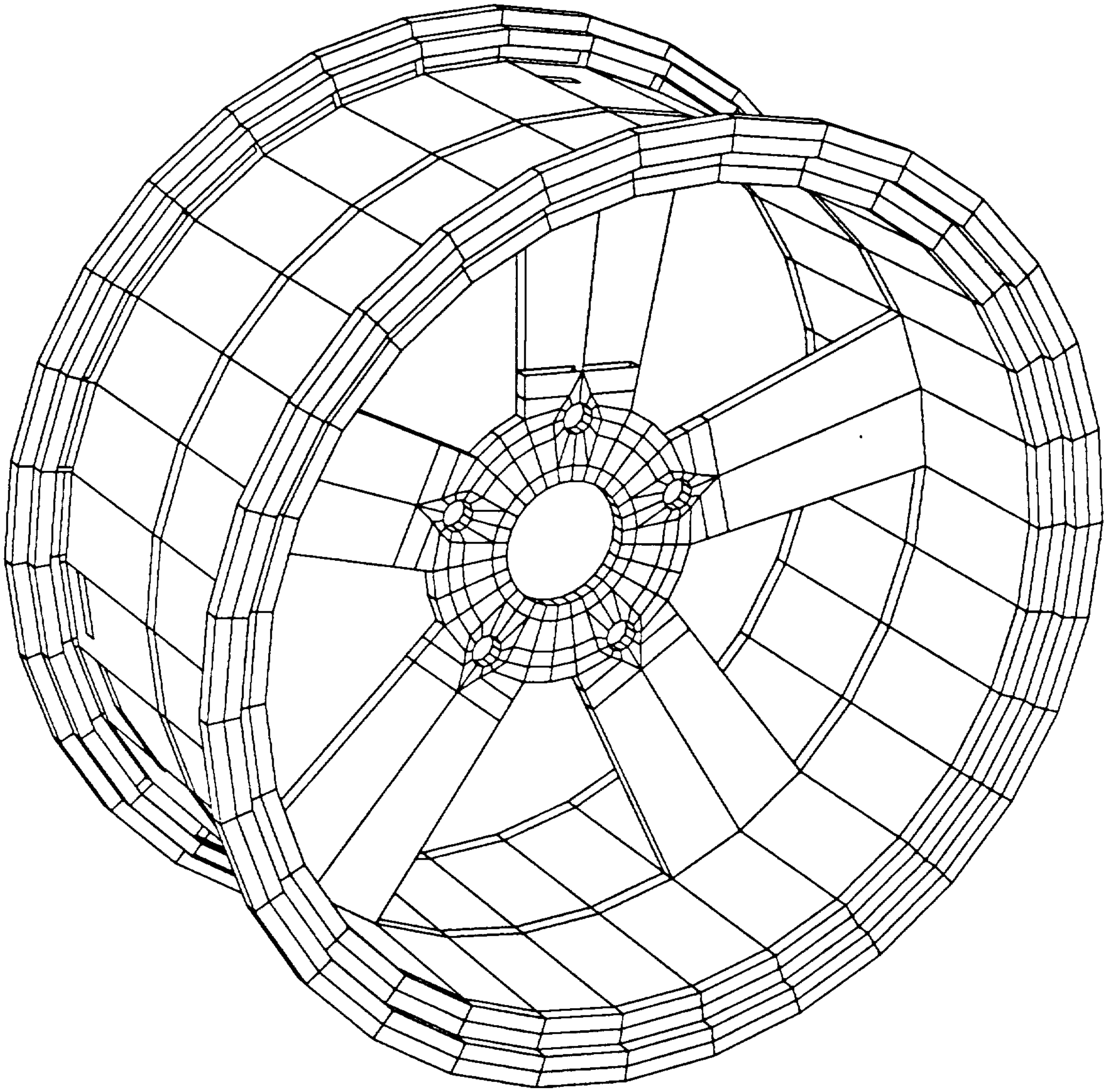
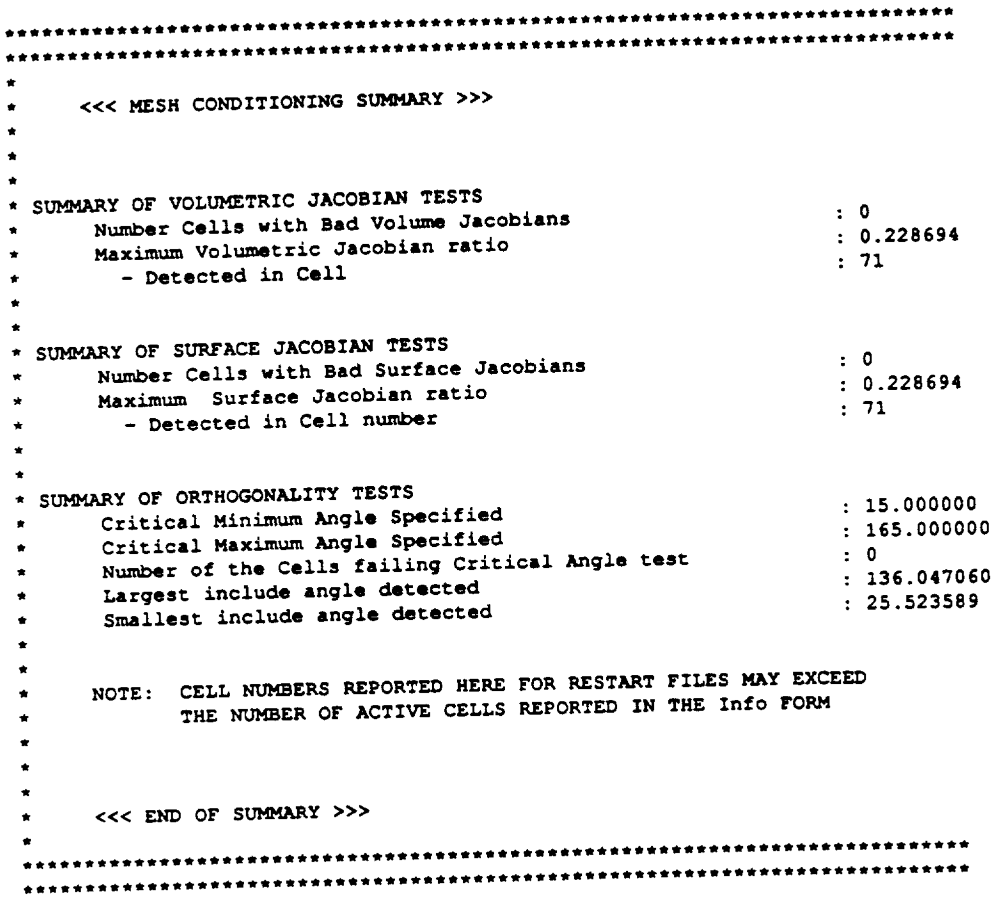


Figure 5.19: Linear finite element used to model three-dimensional wheel rim on a Pontiac Fiero



```
*************************************************************************
*************************************************************************
*
*     <<< MESH CONDITIONING SUMMARY >>>
*
*
*
* SUMMARY OF VOLUMETRIC JACOBIAN TESTS
*       Number Cells with Bad Volume Jacobians                    : 0
*       Maximum Volumetric Jacobian ratio                         : 0.228694
*         - Detected in Cell                                      : 71
*
*
* SUMMARY OF SURFACE JACOBIAN TESTS
*       Number Cells with Bad Surface Jacobians                   : 0
*       Maximum  Surface Jacobian ratio                           : 0.228694
*         - Detected in Cell number                               : 71
*
*
* SUMMARY OF ORTHOGONALITY TESTS
*       Critical Minimum Angle Specified                          : 15.000000
*       Critical Maximum Angle Specified                          : 165.000000
*       Number of the Cells failing Critical Angle test           : 0
*       Largest include angle detected                            : 136.047060
*       Smallest include angle detected                           : 25.523589
*
*
*      NOTE:  CELL NUMBERS REPORTED HERE FOR RESTART FILES MAY EXCEED
*             THE NUMBER OF ACTIVE CELLS REPORTED IN THE Info FORM
*
*
*
*     <<< END OF SUMMARY >>>
*
*************************************************************************
*************************************************************************
```

Figure 5.20: Preprocessing diagnostics for a three-dimensional wheel rim.

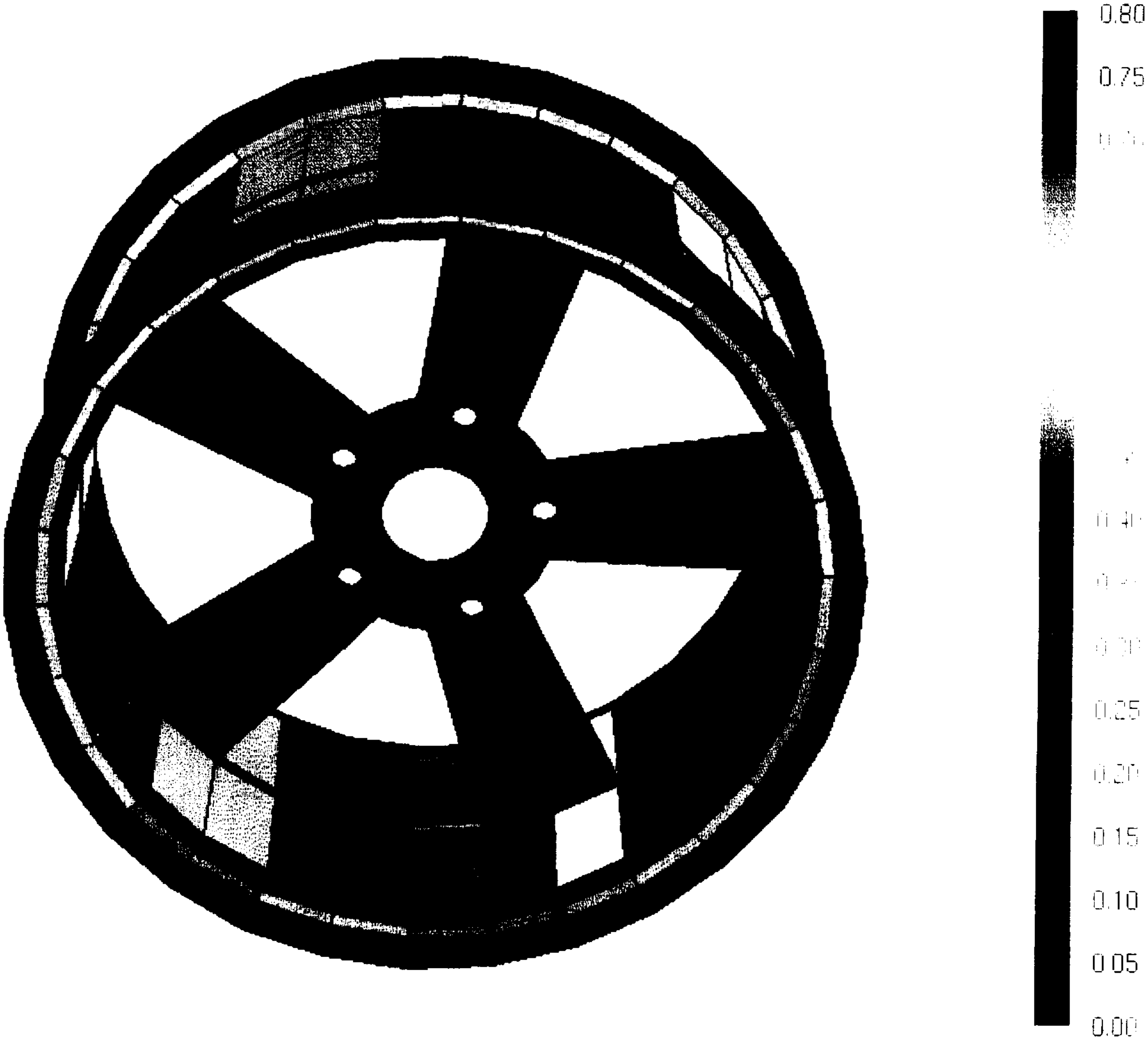


Figure 5.21: Residual error distibution projected on the surface for a linear elasticity analysis of a three-dimensional wheel rim

C-3

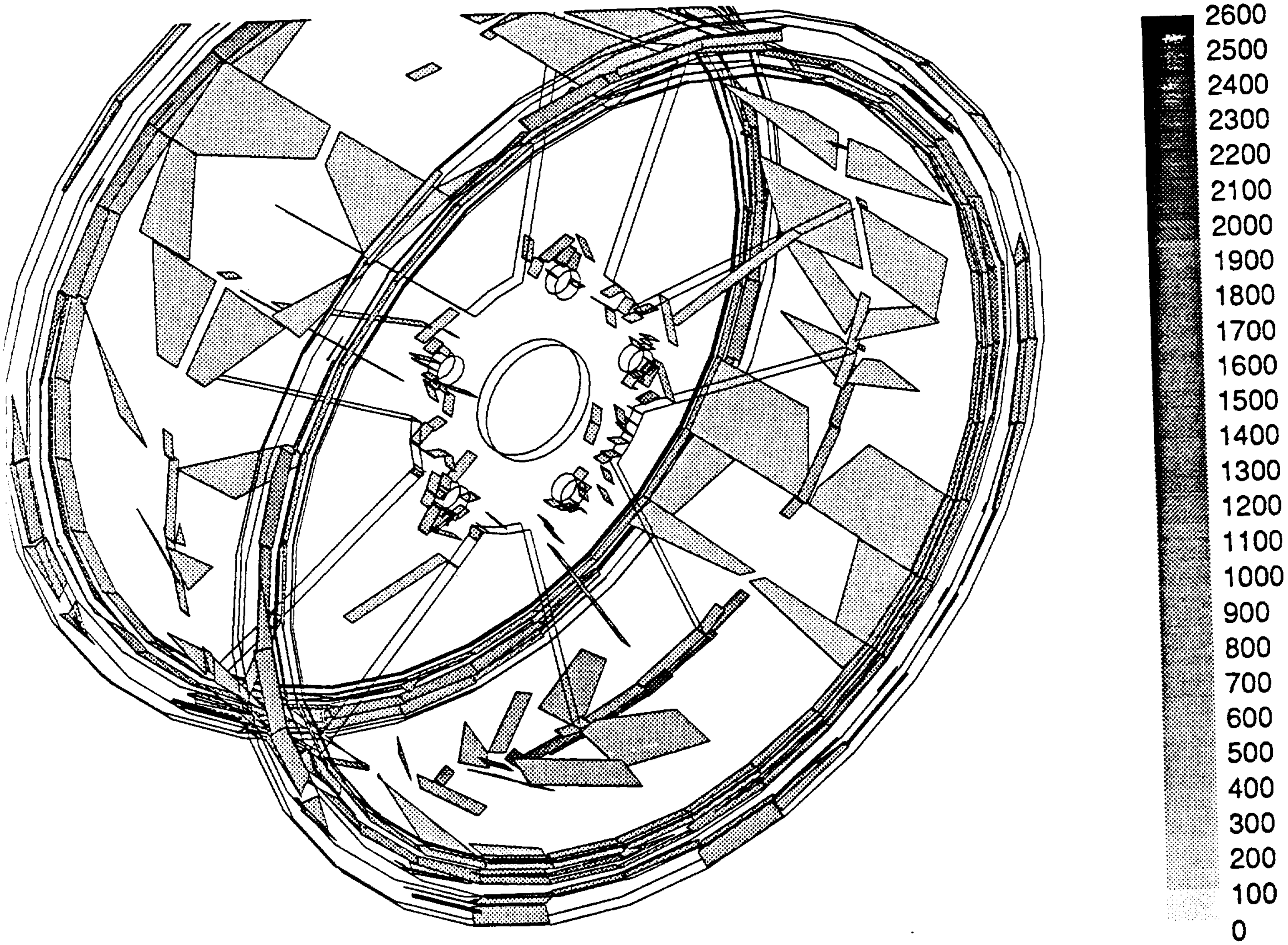


Figure 5.22: Discontinuous Von Mises stress isosurfaces for the three-dimensional wheel rim

5.4 Incompressible Viscous Flow

5.4.1 Backstep Channel (TEACH - finite difference solution)

The first incompressible viscous flow problem presented herein is a two-dimensional backstep channel problem. The geometry and boundary conditions used in the TEACH code simulation are shown in Figure 5.24. Summarizing the boundary conditions in this figure, there is a prescribed quadratic velocity profile along the upper half of the left hand face given by

$$u(x,y,z) = 1.5 * [1.0 - ((y - 1.5)/0.5) * *2],$$

an outflow boundary along the right face with

$$\begin{aligned} p &= 0.0, \\ \frac{du}{dx} &= \frac{dv}{dx} = 0.0, \end{aligned} \tag{230}$$

and solid wall boundaries elsewhere. The Reynolds number used in the solution process was set to 100 based on the height of the channel.

Numerical Model

The geometry, shown in Figure 5.24, has initially discretized into a 35 x 31 finite difference mesh which has been clustered in horizontal direction near the inflow region and around the separation point. The distribution of nodal points in the vertical direction is uniform with the first point occurring at a distance of 0.0667 from the solid wall, see figure 5.25. Note that this geometry the entrance length up to the backstep has been removed and a fully developed quadratic profile has been imposed to simulate the upstream effects.

AUDITOR Analysis

The two-dimensional grid used during the solution phase is again a very simple rectangular grid which has some moderate clustering, and thus there is no need to preprocess the mesh. The features of interest include, the recirculation zone in the lower left portion of the mesh, the location of the separation point, and the distribution of the error in the numerical solution. To ensure that the numerical solution has been correctly read and placed in the database, a plot of the isosurfaces for the u-velocity and pressure are shown in Figures 5.26 and 5.27. From a plot of the u-velocity for a value of -0.004, and with the aid of the graphical picker, the separation point for this model is estimated to be 2.875 which corresponds very well with numerically accepted value of 3.0.

Activating the residual error estimator, the estimated global error in the solution is 117% with a maximum local error of 24.6%. Figures 5.28 and 5.29 show the corresponding

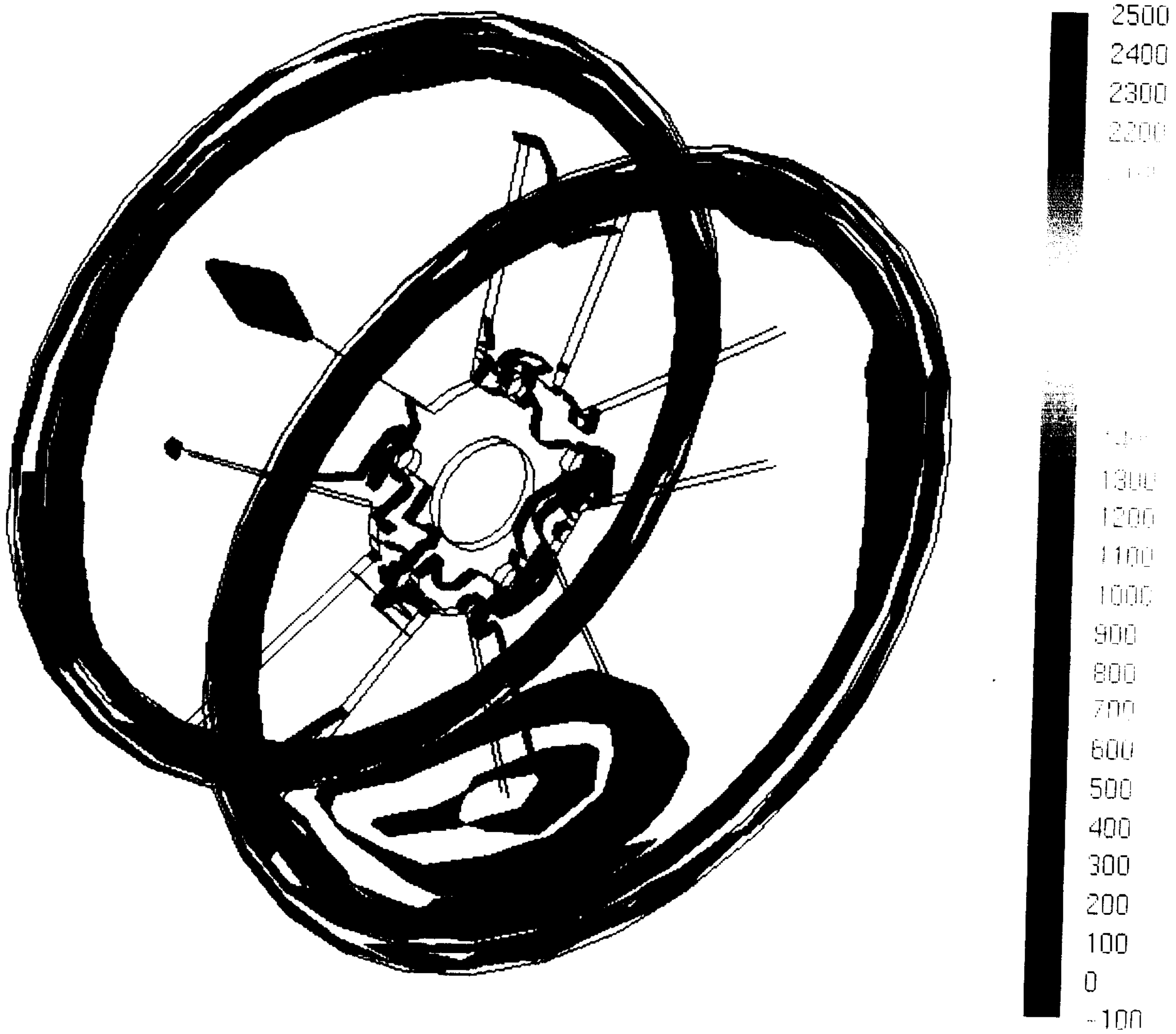


Figure 5.23: Enhanced Von Mises stress isosurfaces for a three-dimensional wheel rim

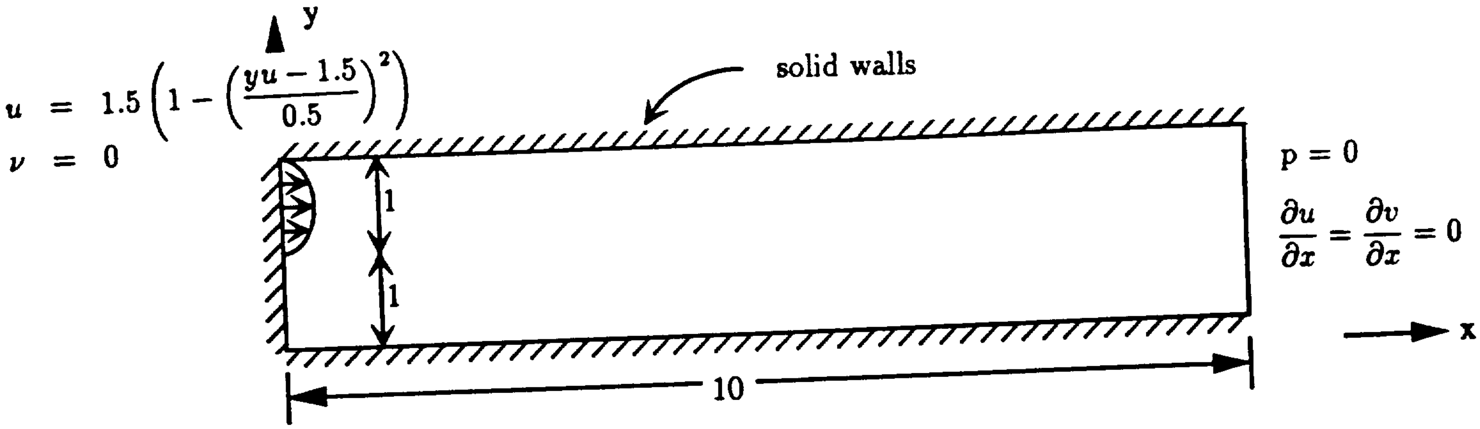

y
$u = 1.5\left(1 - \left(\frac{yu - 1.5}{0.5}\right)^2\right)$
$\nu = 0$
solid walls
1
1
p = 0
$\frac{\partial u}{\partial x} = \frac{\partial v}{\partial x} = 0$
x
10

Figure 5.24: Backstep geometry used with the TEACH flow solver, Re=100.

distribution of the residual error projected on the surface and a zoom of the error near the inflow region. Based on the coloring in Figure 5.28 the largest cell-wise error are all occurring at or near the inflow boundary. The zoom of the errors in this region, rescaled in Figure 5.29 to a maximum error of 0.10, shows additional relatively large errors occurring directly downstream of the inflow region and bounding the recirculation zone.

The final phase in the analysis, consists of accessing the postprocessing option to extract higher order results for the vorticity magnitude. For reference, the current distribution of the vorticity magnitude have been calculated as shown in Figure 5.30. Note, that the vorticities are highly discontinuous and the isosurface window indicates that the maximum vorticity is 6.75. Figure 5.31 shows the corresponding vorticity distribution after the solution enhancement has been activated. Note, from the isosurface window the vorticity magnitude now has a maximum vorticity magnitude of 7.07 and small undershoot as well.

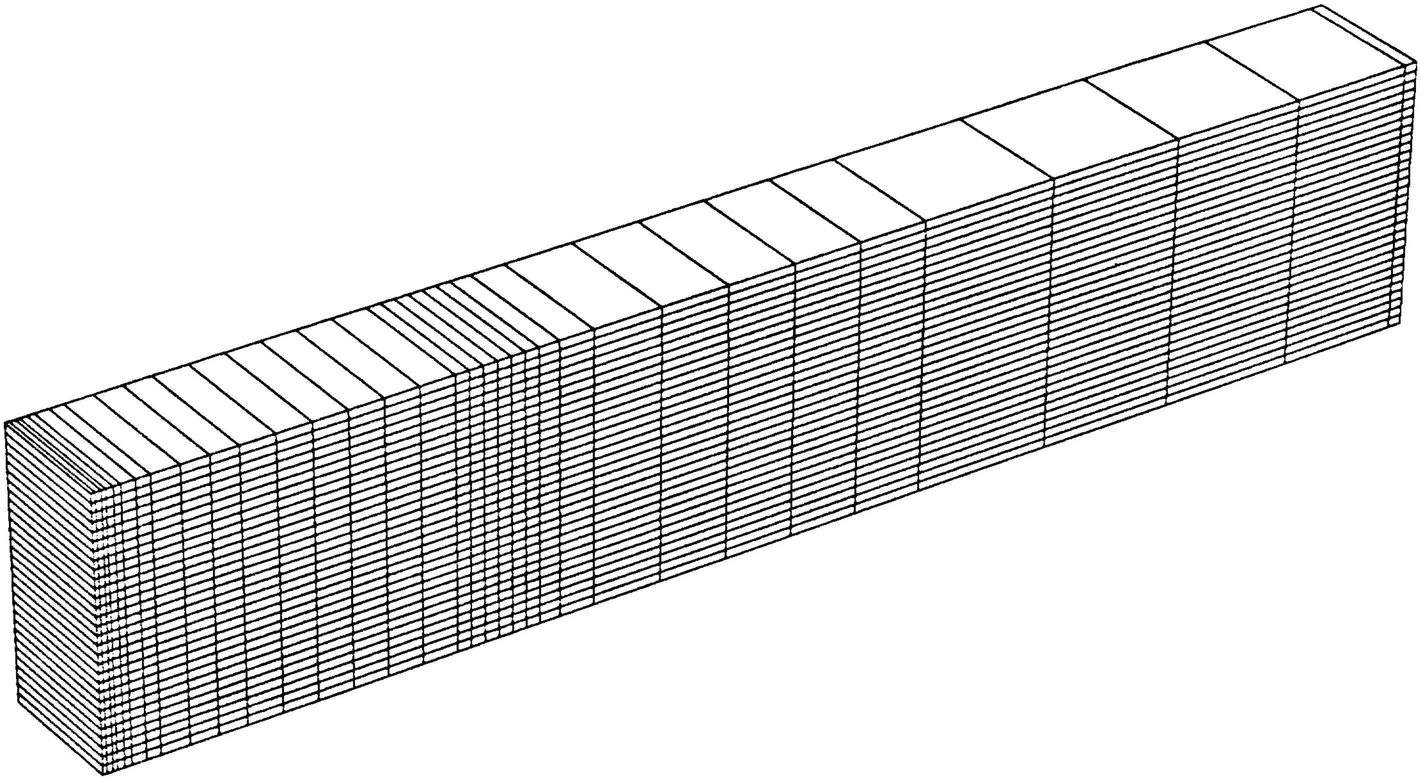


Figure 5.25: Computational mesh used in the TEACH code simulation and *AUDITOR* analysis of a two-dimensional backstep, Re = 100

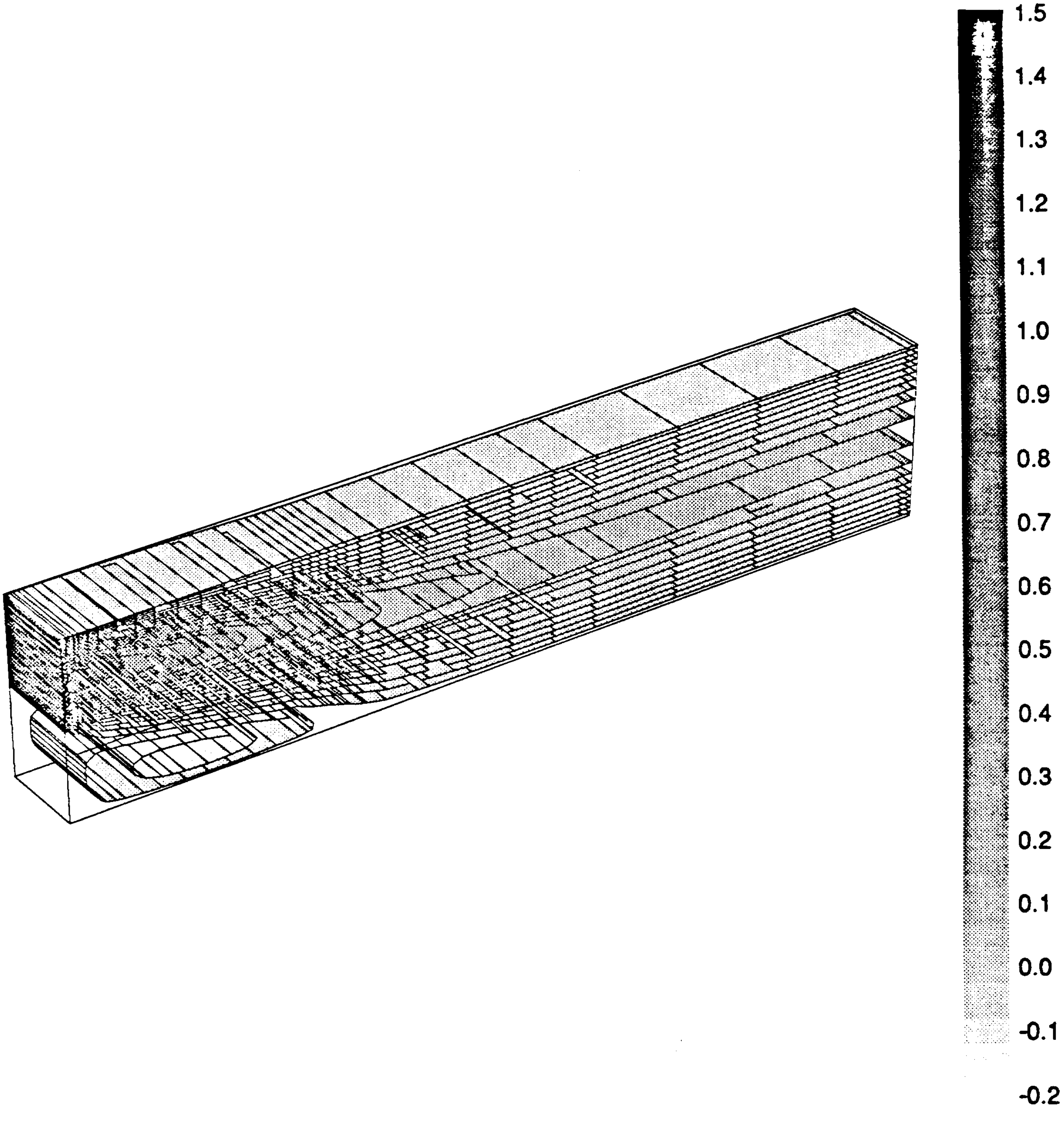


Figure 5.26: U-velocity isosurfaces for the two-dimensional backstep, Re = 100

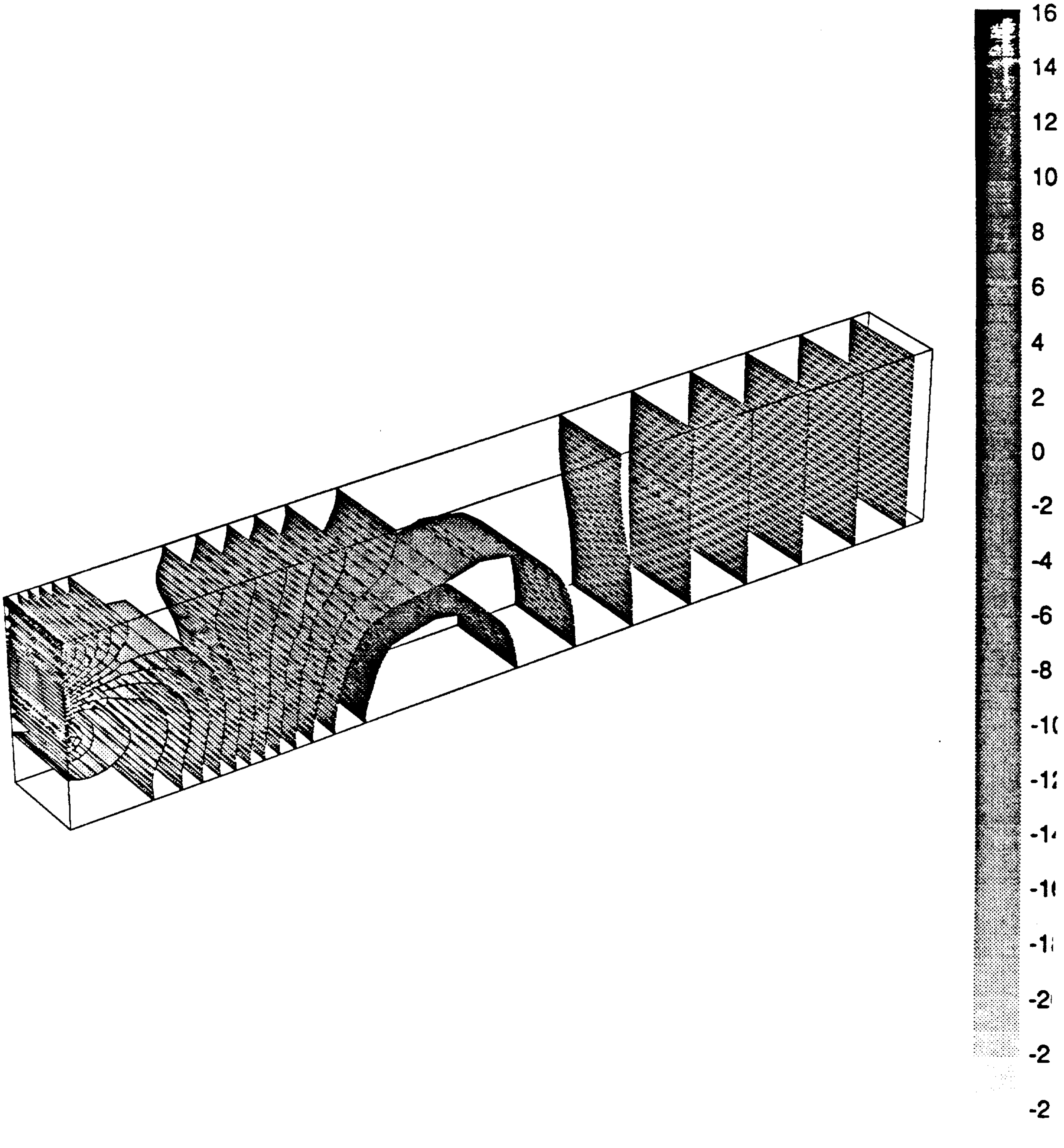


Figure 5.27: Pressure isosurfaces for the two-dimensional backstep, Re = 100

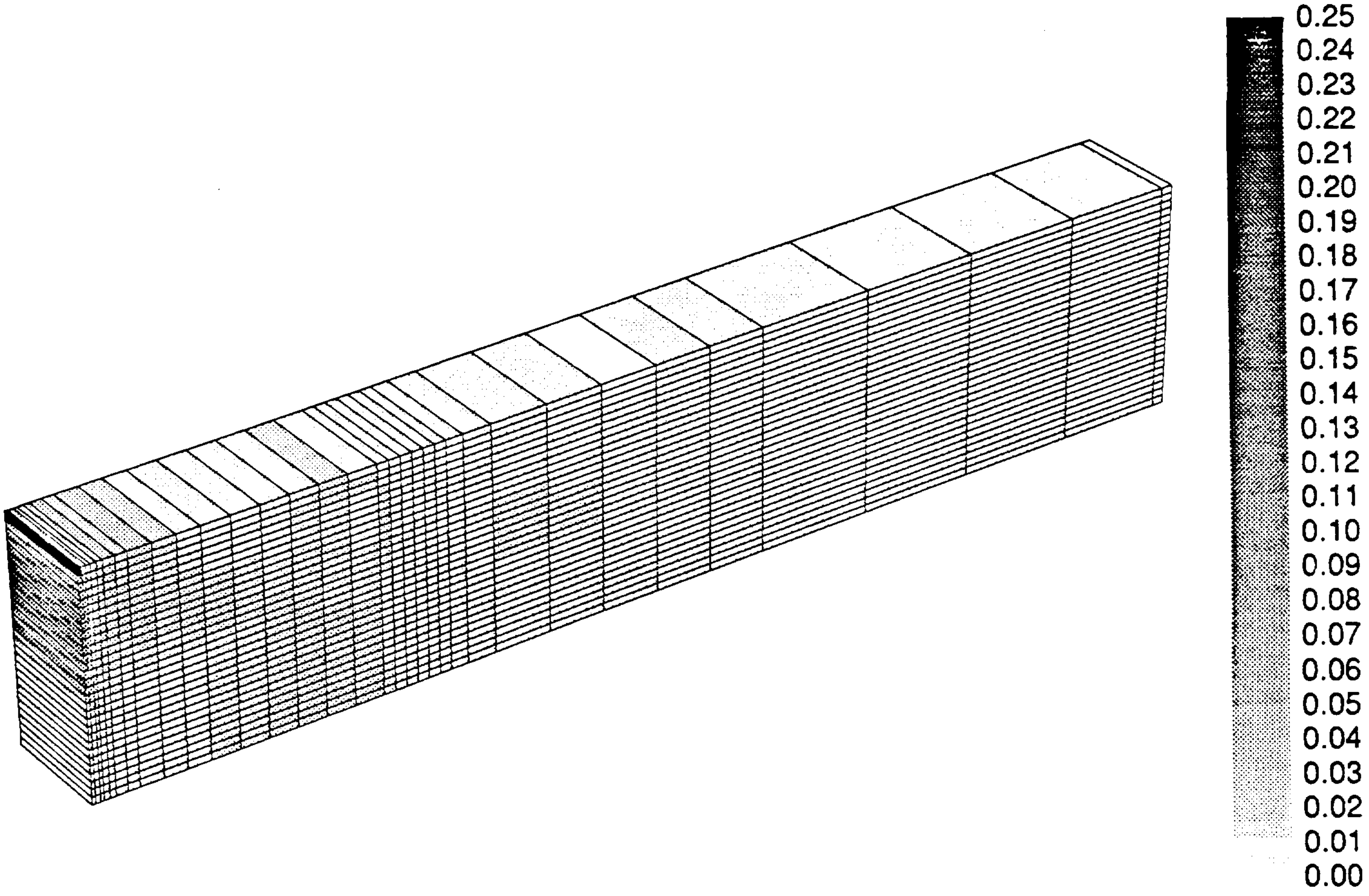


Figure 5.28: Residual error distribution for the two-dimensional backstep, Re = 100

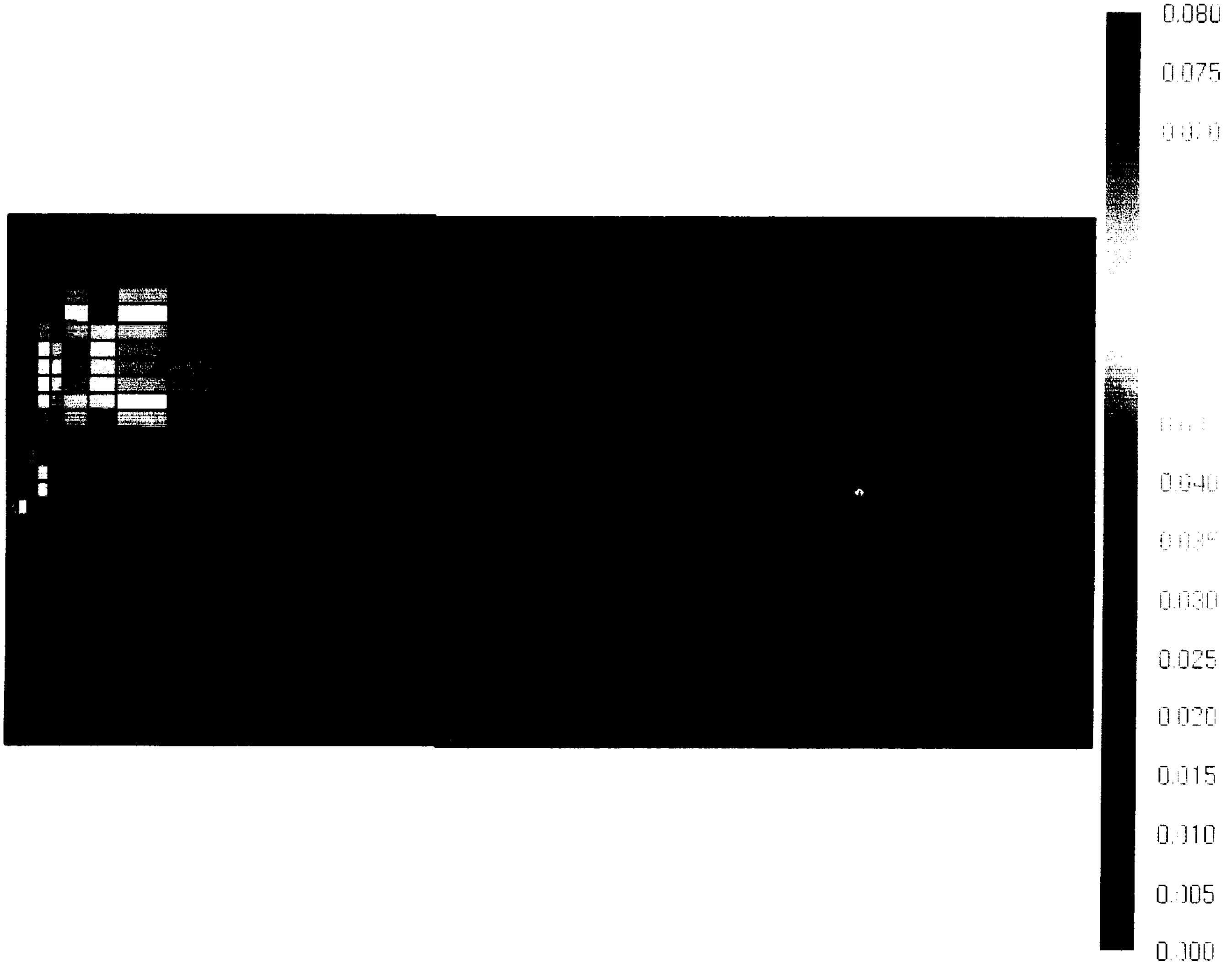


Figure 5.29: Zoom of the residual error near the inflow for the two-dimensional backstep, Re = 100

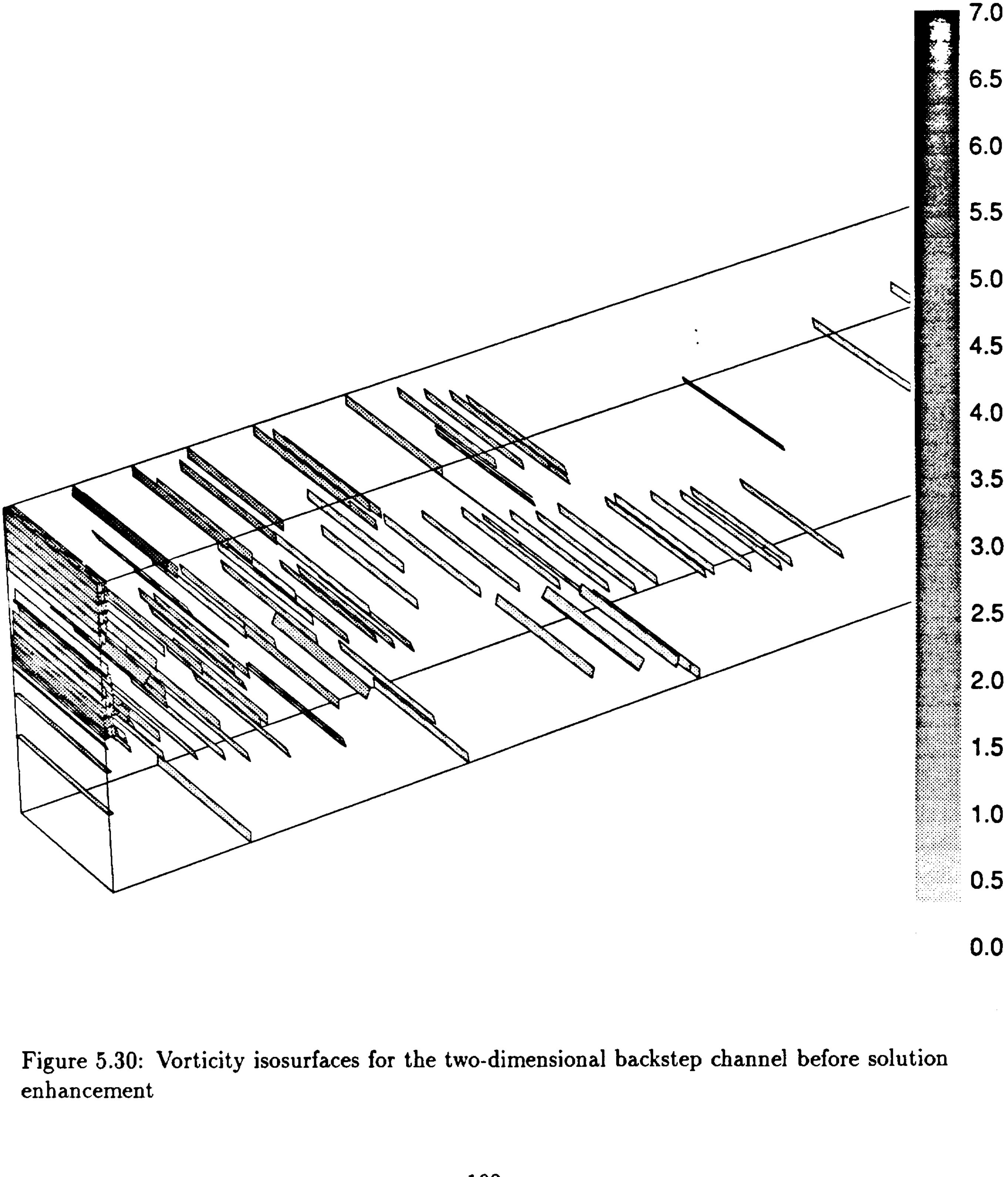


Figure 5.30: Vorticity isosurfaces for the two-dimensional backstep channel before solution enhancement

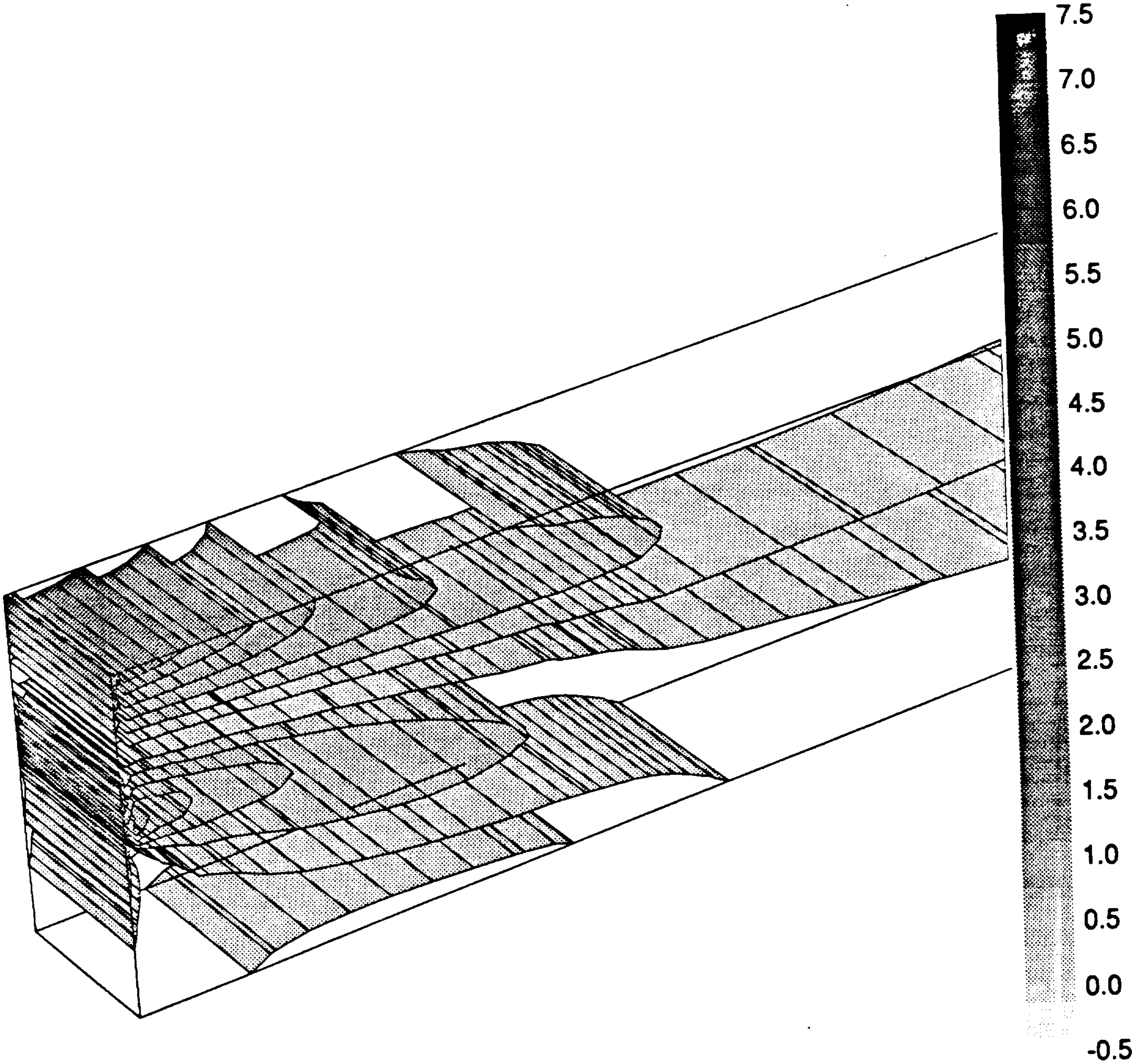


Figure 5.31: A sketch of the flow domain for the flow through a square-duct with a bend, Re = 790

5.4.2 90° Bend (P3/CFD - finite element solution)

The final incompressible viscous flow problem presented herein is a three-dimensional flow through a rectangular duct with a 90° bend. The geometry and boundary conditions used in the P3/CFD analysis are shown in Figure 5.32. Summarizing the boundary conditions shown in this figure, there is a prescribed uniform velocity profile along the vertical face on the far right,

$$\begin{aligned} u &= -1.0 \\ v &= 0.0 \\ w &= 0.0 \end{aligned} \tag{231}$$

an outflow boundary on the horizontal face at the top with

$$p = 0.0 \tag{232}$$

and solid wall boundaries elsewhere. The Reynolds number used in the solution process was set to 790 based on the height of the duct.

Numerical Model

The geometry for this example, shown in Figure 5.32, has been discretized into 28 quadratic finite elements with 375 degrees of freedom. In this model, there are 7 elements along the length of the duct which have been clustered in the bend region and four elements in the cross section, see Figure 5.33. Note, this is a very coarse model for this problem, and that the goal here is not to obtain a highly accurate solution from the solver but estimate the error in the numerical solution.

AUDITOR Analysis

The three-dimensional grid used during the solution phase is again a very simple rectangular grid, which has some moderate clustering, and thus there is basically no need to preprocess the mesh. The features of interest include the development of vortex tubes in the bend region of the duct, a possible recirculation zone at the end of the bend, and the distribution of the error in the numerical solution. To ensure that the solution has been correctly read and placed in the database, a cross section plot of the velocity magnitude and isosurfaces of the pressure are shown in Figures 5.34 and 5.35. For reference, we have also plotted the z-velocity components, see Figure 5.36, which shows the out of plane recirculation in the bend region.

Activating the residual error estimator, the estimated global error in the solution is 778% with a maximum local error of 256%. Figure 5.37 shows a zoom of the corresponding distribution of the residual error projected on the surface of the duct and Figure 5.38 shows the error projected onto a planar slice through the center of the duct. Based on the gray

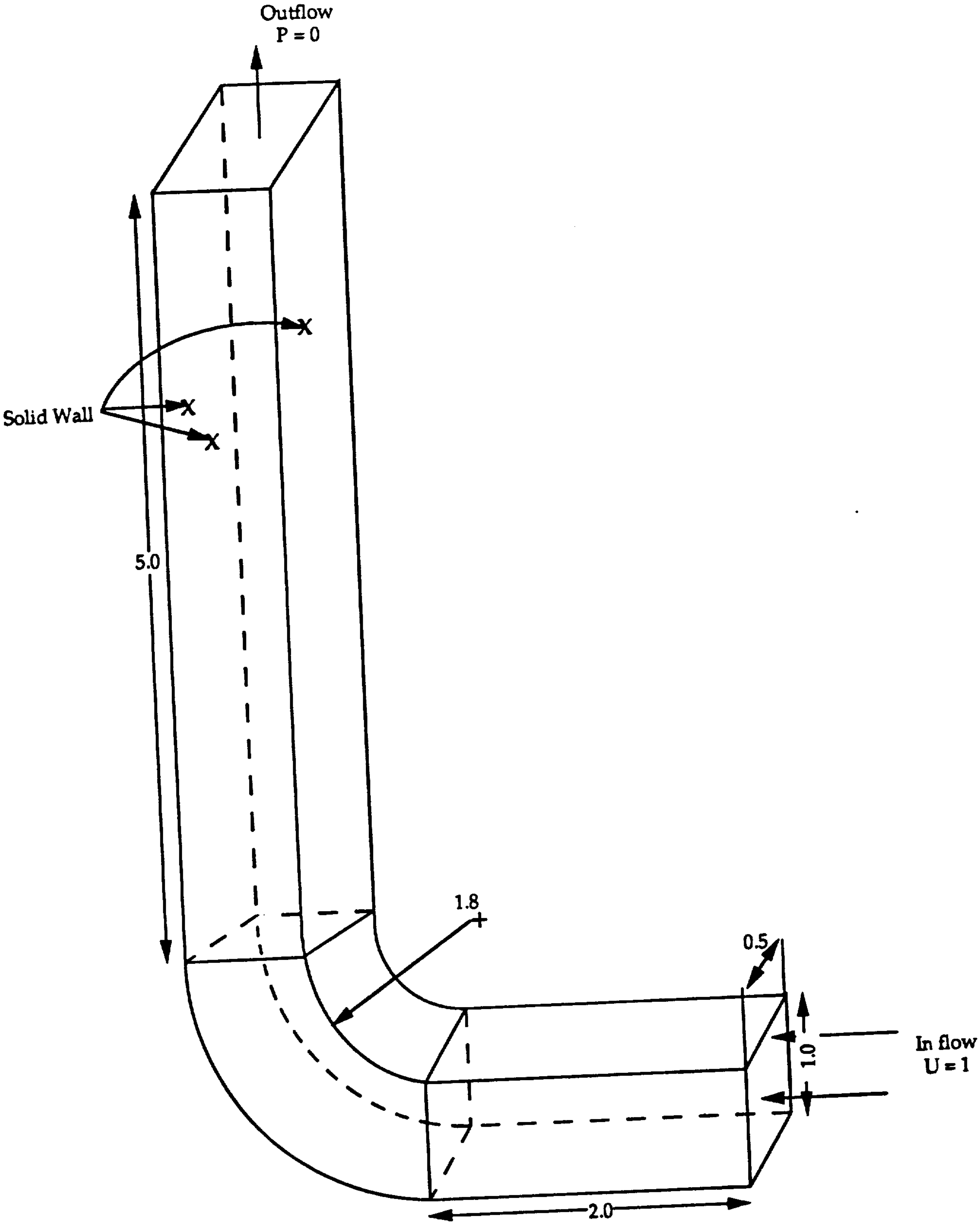


Figure 5.32: Vorticity isosurfaces for the two-dimensional backstep channel after solution enhancement

scale coloring in this figure, the maximum cell-wise errors are occurring at the inflow and the outflow sections of the domain. The error at the inflow is most likely a result of the technique which is being used to enforce uniform inflow velocity profile. The relatively large error at the outflow section is at least in part due to the relatively large size of the quadratic elements used near the outflow boundary. A finer mesh in either of these regions would most probably reduce the errors there significantly. Finally, note that due to the coarseness of the finite element model there is not separation of the flow at the exit point of the bend, however, there still appears to be relatively large local errors in this region.

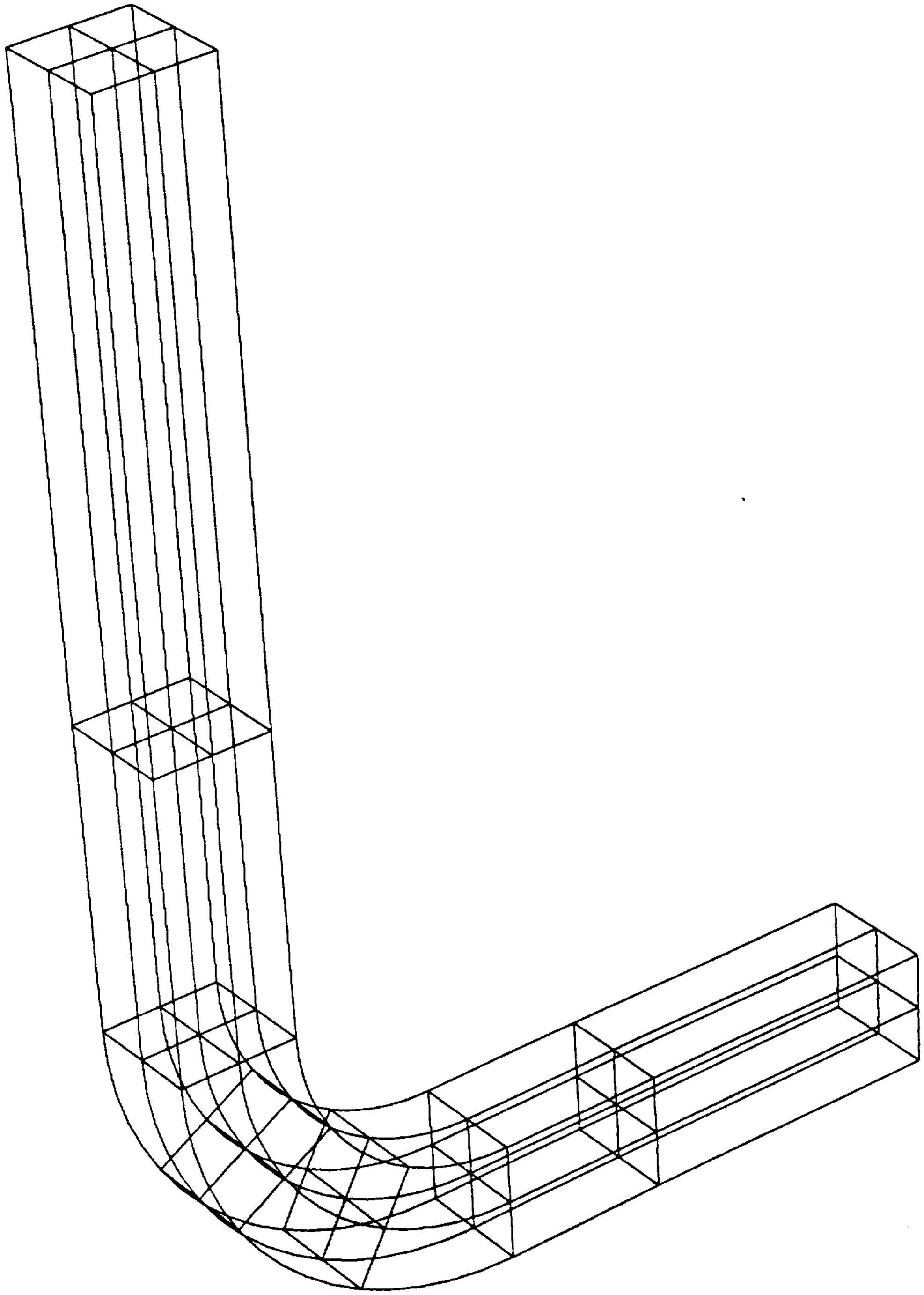


Figure 5.33: Quadratic computational mesh used in the 90° bend analysis, Re = 790

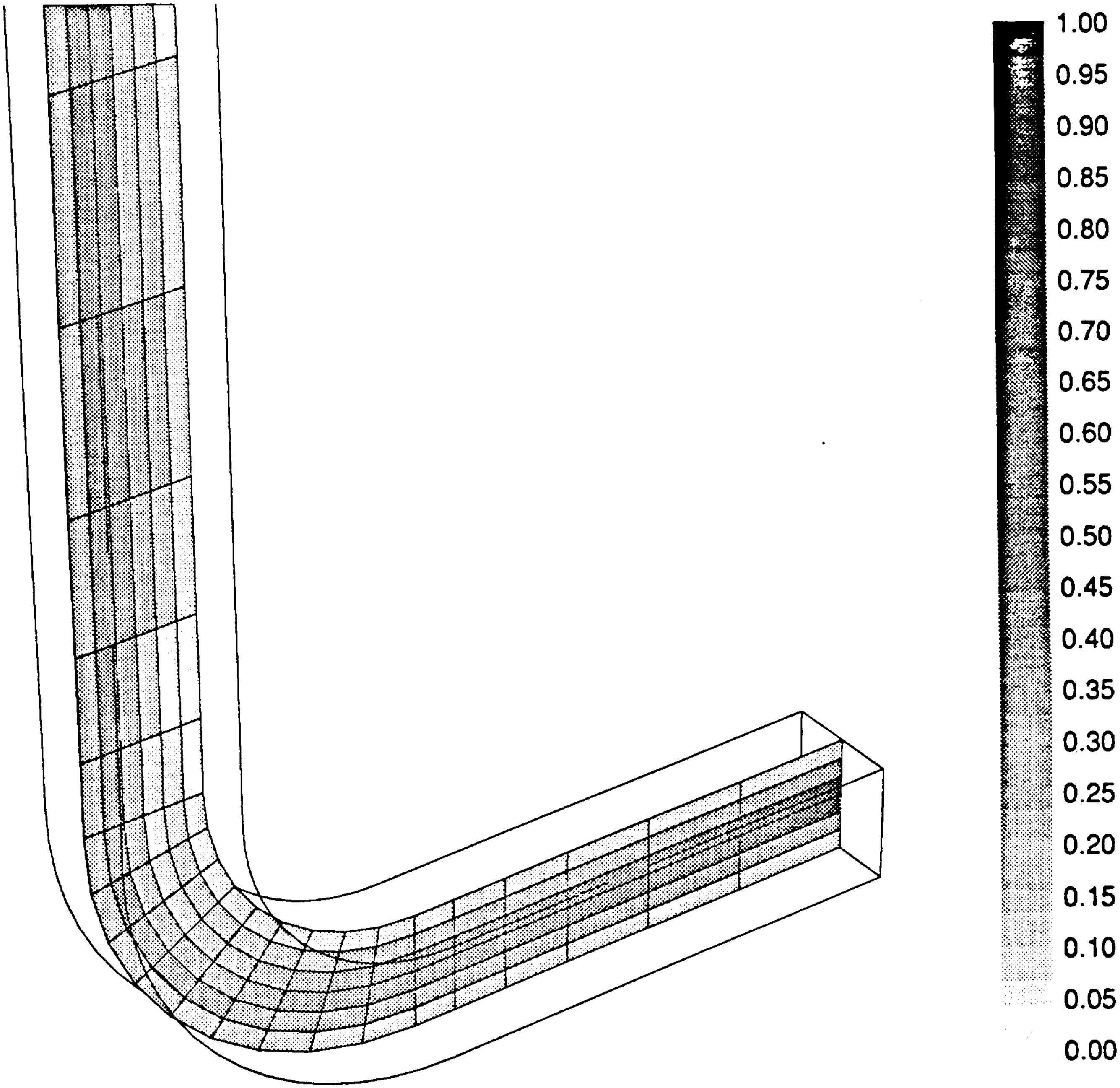


Figure 5.34: Planar slice showing the velocity magnitude at the centerline of the 90° bend

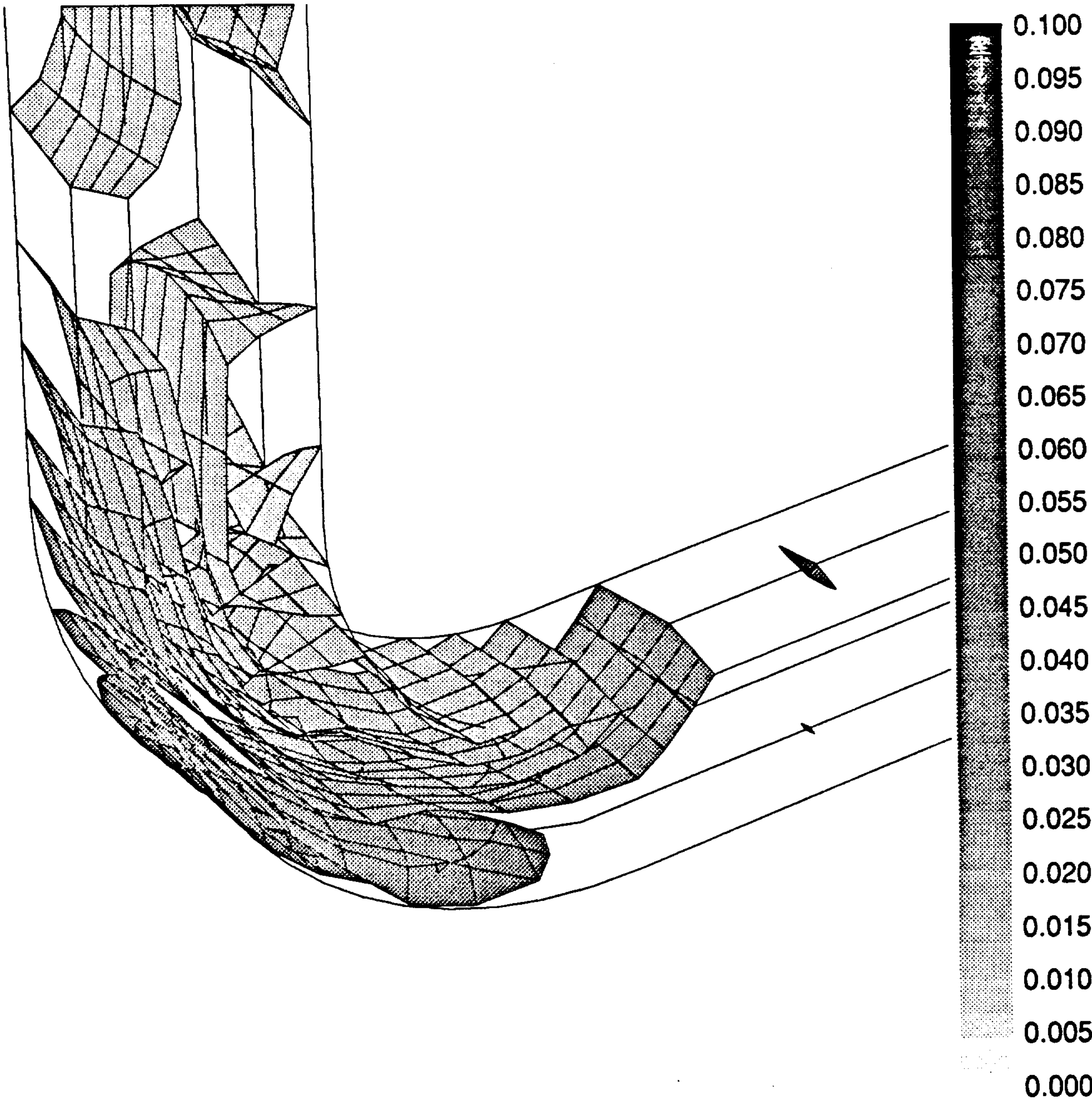


Figure 5.35: Pressure isosurfaces in the bend region of the rectangular duct

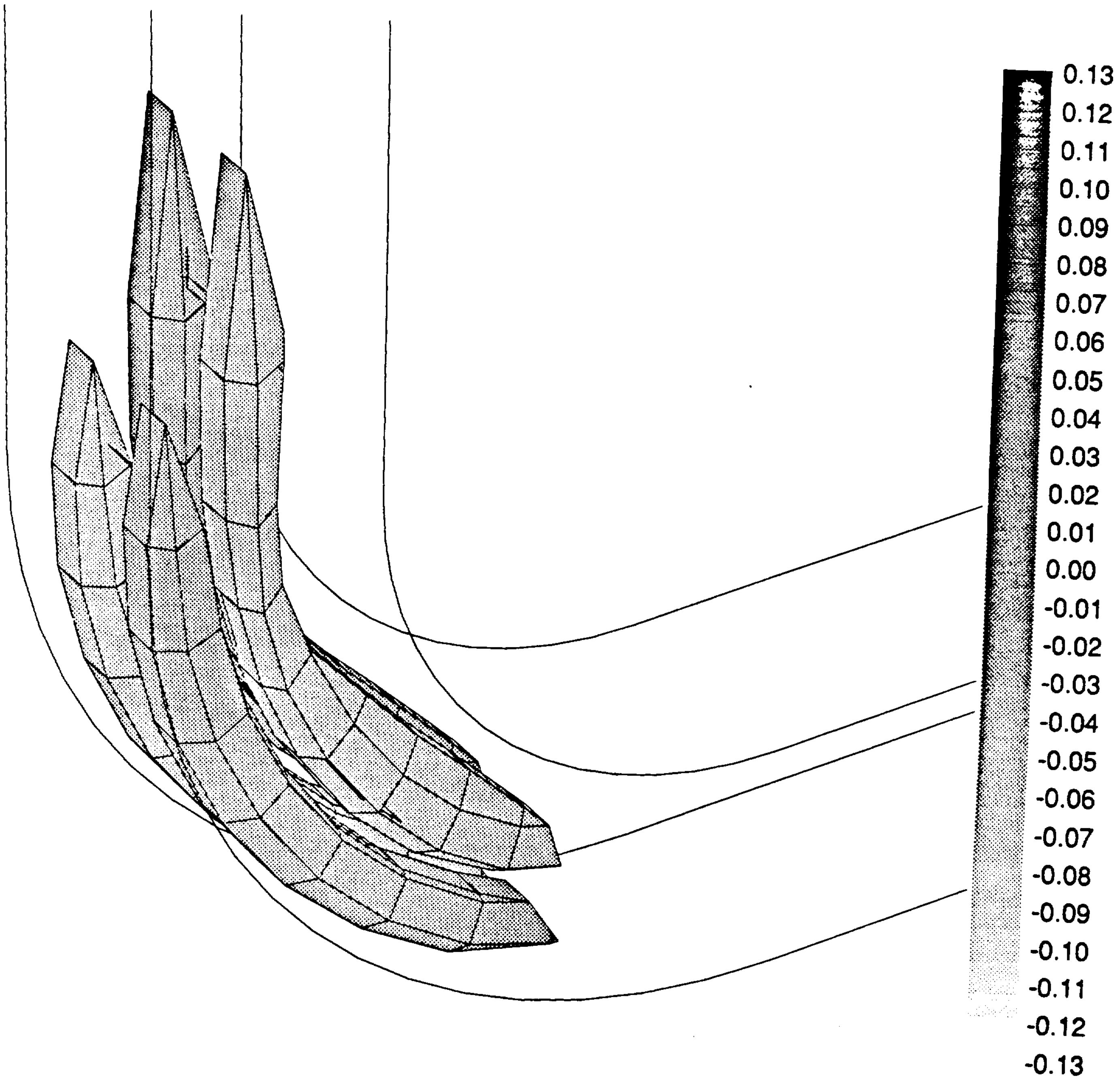


Figure 5.36: Z-velocity isosurfaces in the bend region of the rectangular duct showing three-dimensional recirculation

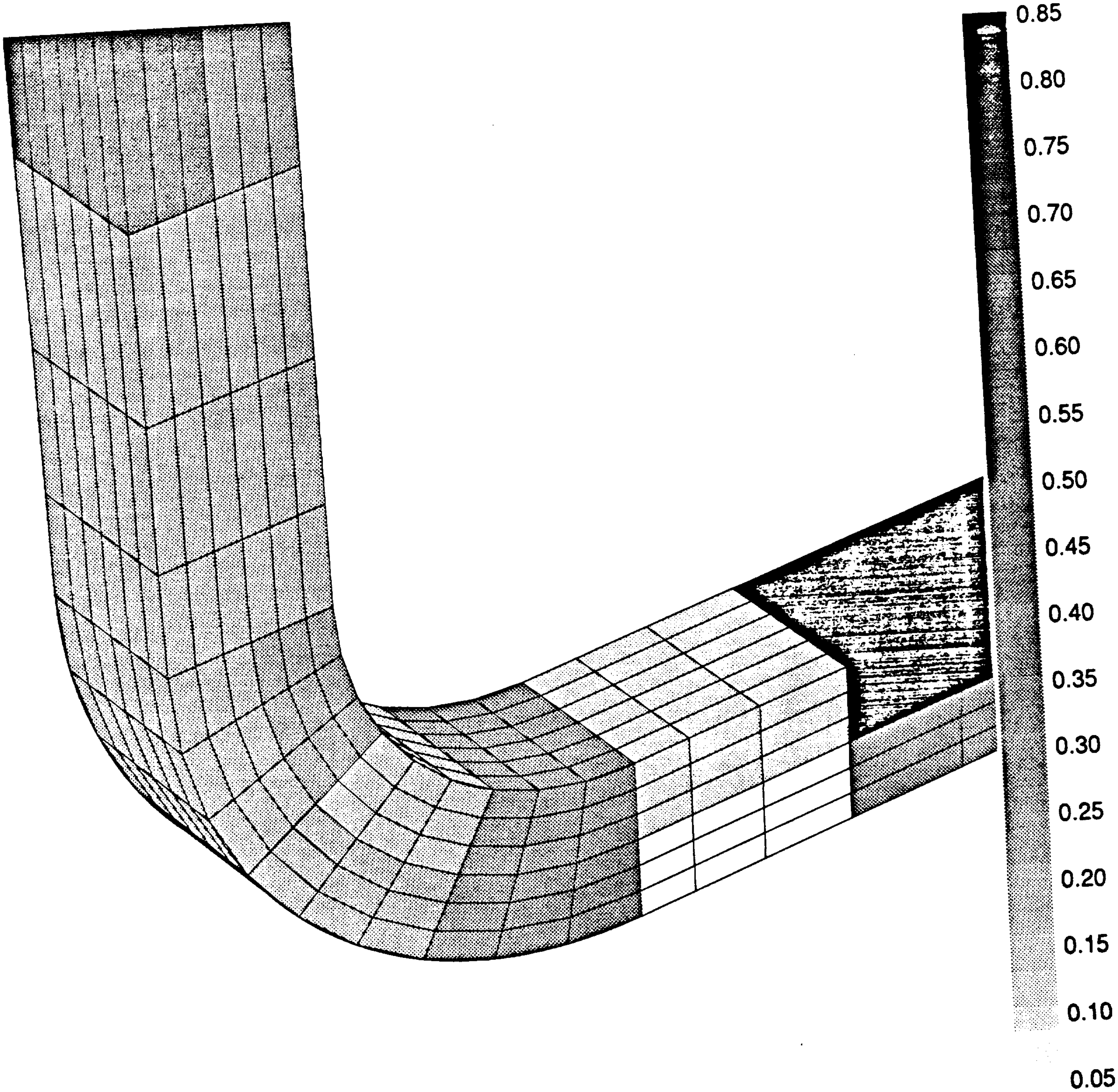


Figure 5.37: Zoom of the residual error distribution in the bend region of the duct projected on the surface

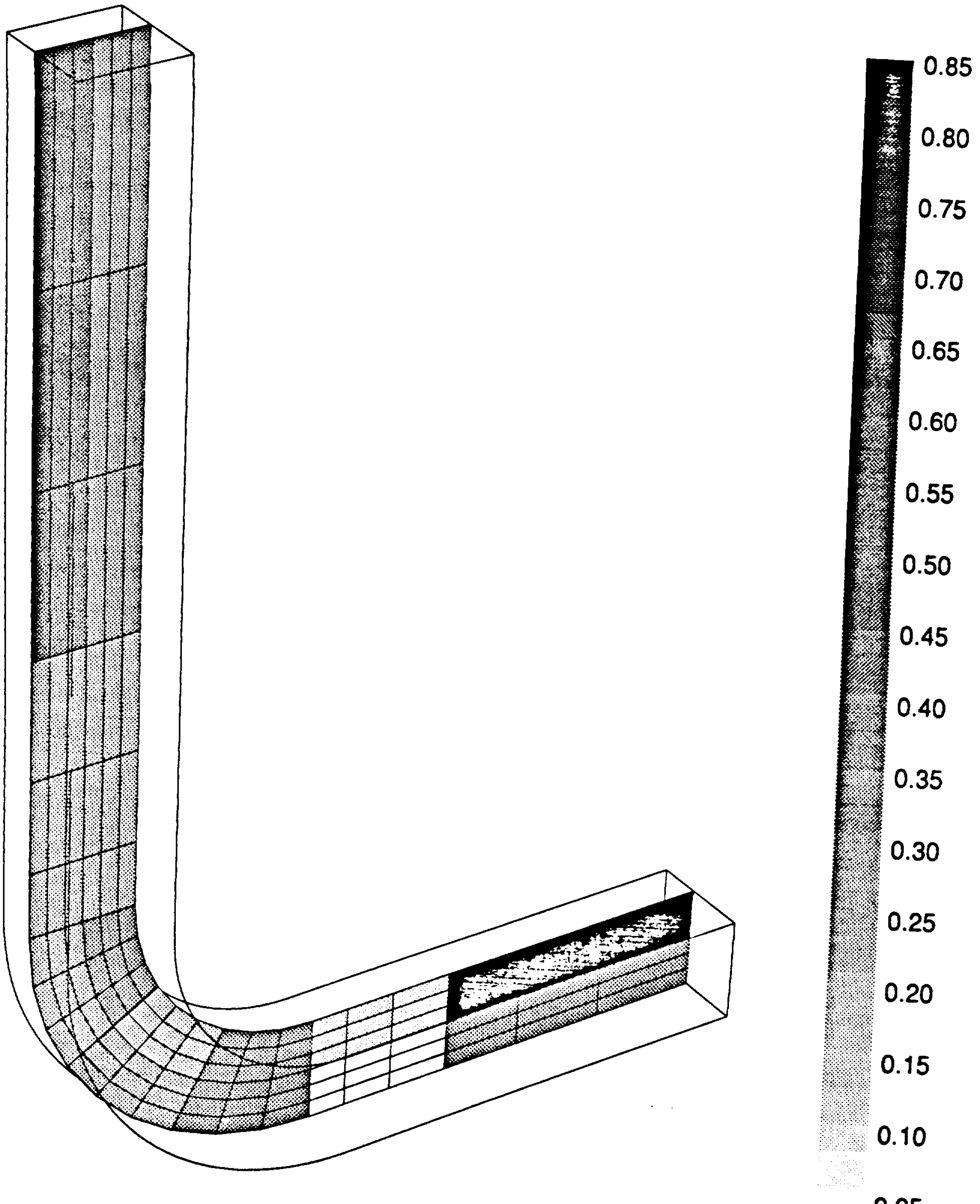


Figure 5.38: Planar slice through the center of the rectangular duct colored by the error distribution

The final phase in the analysis of this problem consists of accessing the postprocessing options to extract higher order results for the vorticity magnitude. Figure 5.39 shows the corresponding vorticity magnitude distribution after the postprocessing phase. Note, the scale in this figure shows the enhancement phase has caused an undershoot in the vorticity magnitude.

5.5 Compressible Flow

5.5.1 Airfoil (JCODE - finite volume solution)

The first compressible flow problem presented herein is a two-dimensional inviscid flow past a NACA 0012 airfoil. The geometry and boundary conditions used in the JCODE simulation are shown in Figure 5.40. Summarizing the boundary conditions shown in this figure; far field data are prescribed along the entire outer parameter of 0-grid, and solid wall conditions (no penetration) are specified on airfoil. The far field conditions are

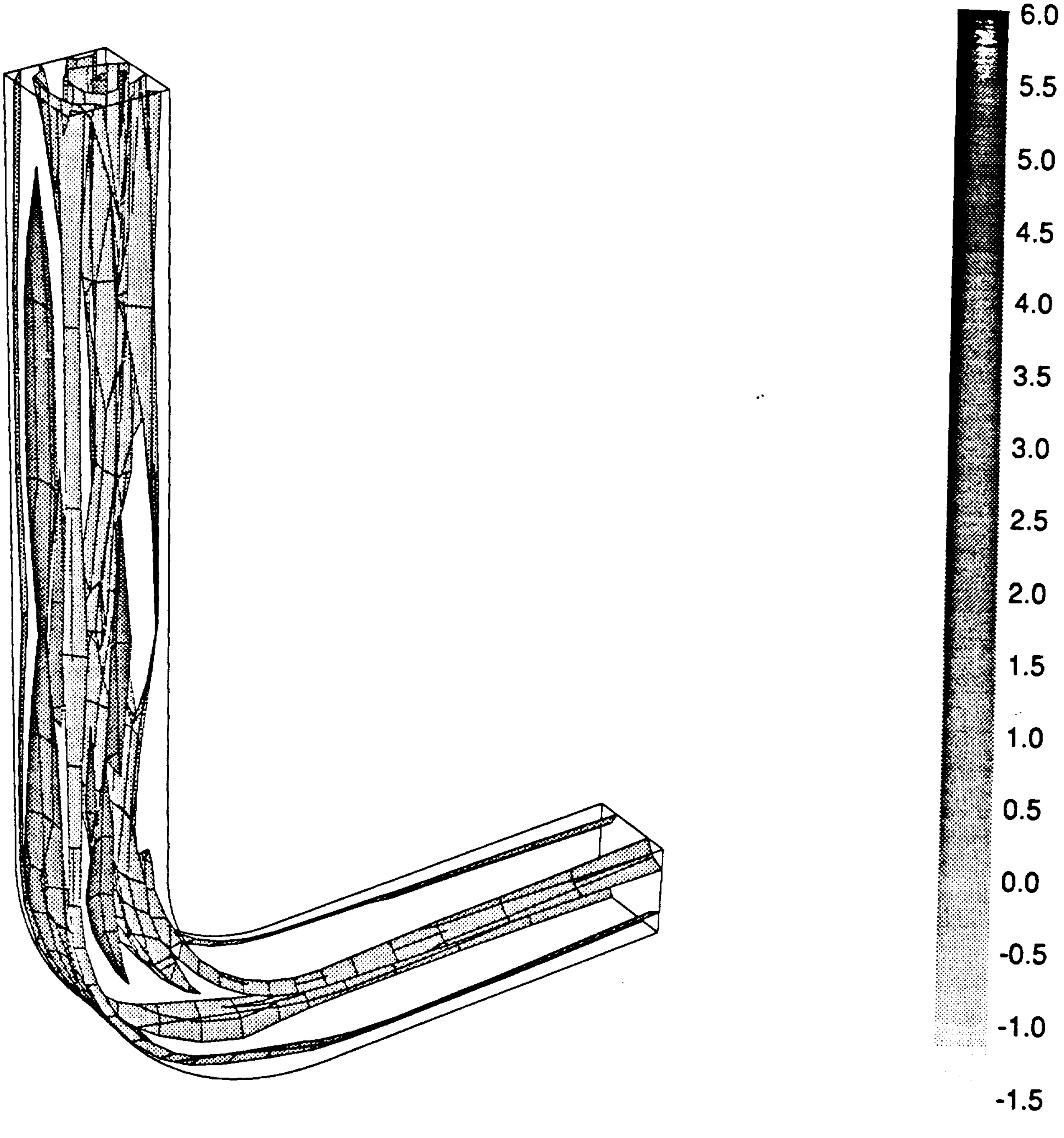


Figure 5.39: Isosurface of the vorticity magnitude after postprocessing

$$
\begin{aligned}
\text{Mach} &= 0.75 \\
\text{alpha} &= 2.0^\circ \\
\text{gamma} &= 1.4
\end{aligned}
$$

All other quantities used in the computations are non-dimensionalized by the density and velocity at infinity.

Numerical Model

The geometry shown in Figure 5.40, has been discretized into 1200 cells which are clustered near the airfoil in the radial direction and near the leading and trailing edge in the primary flow direction. A zoom of the mesh near the airfoil is shown in Figure 5.41.

AUDITOR Analysis

The first phase of the *AUDITOR* analysis for this test problem accesses preprocessor to gather diagnostics concerning the mesh quality. Figure 5.42 shows a copy of the diagnostics which appear on the screen as of result of this preprocessing phase. Of note here are the following: there are no elements with bad surface or volume jacobians, the largest and smallest included angles in the mesh are 171.9 and14.4 respectively, and the largest jacobian ratio was found the be 0.175.

The flow features of interest for this case are the Mach number distribution and reattachment point of the shock on the upper surface, the pressure distribution along the surface of the airfoil, and the distribution of the error in the numerical solution. To ensure that the numerical solution has been correctly read and placed in the database, a plot of the isosurfaces for the Mach number and pressures are shown in Figures 5.43 and 5.44. Based on the data contained in Figure 5.43, and the isosurface editor form, the maximum Mach number predicted by the JCODE is 1.29 which compares well with other results reported in the literature. For this case, we have also plotted the pressure distribution along the surface of the wing, see Figure 5.45.

Activating the residual error estimator, the estimated global error in the solution is 0.29% with a maximum local error of 0.106%. Figure 5.46 shows a zoom of the residual error distribution near the airfoil. Based on the color scale in this figure, the maximum cell-wise errors are occurring near the leading edge of the airfoil on both the section and pressure surfaces and at the shock reattachment point near the center of the airfoil.

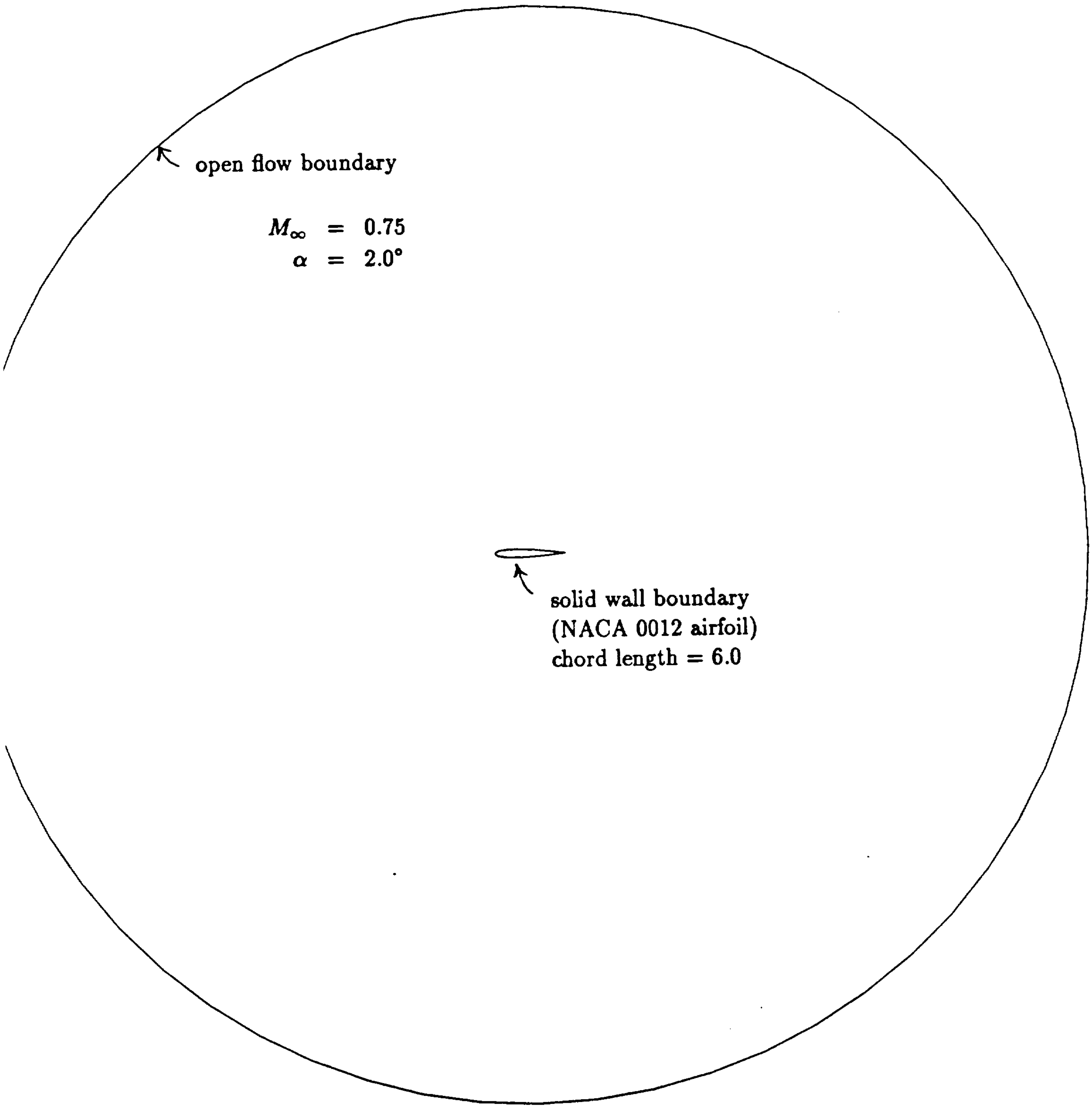


Figure 5.40: Geometry and boundary conditions for inviscid flow over a two-dimensional airfoil

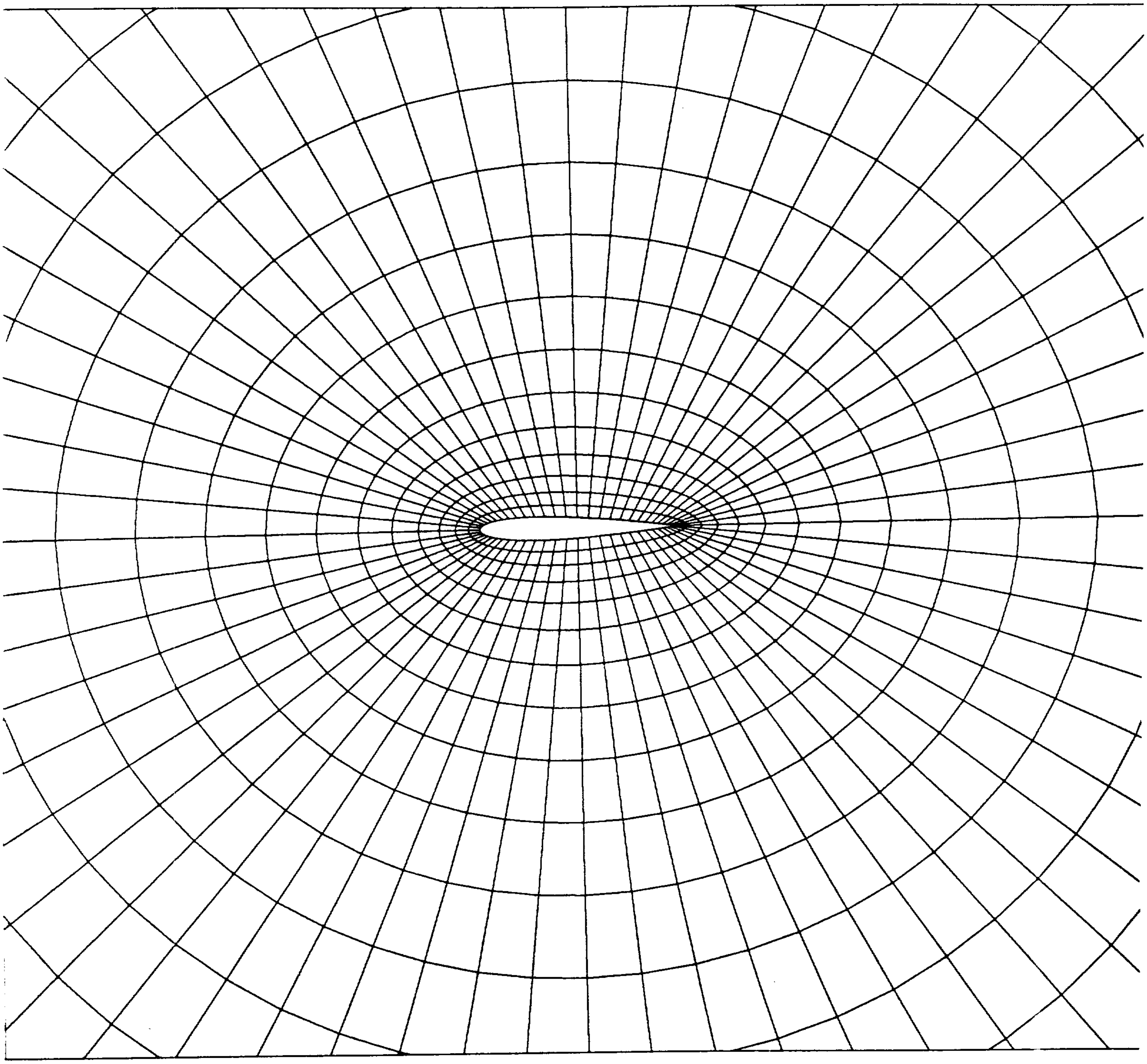
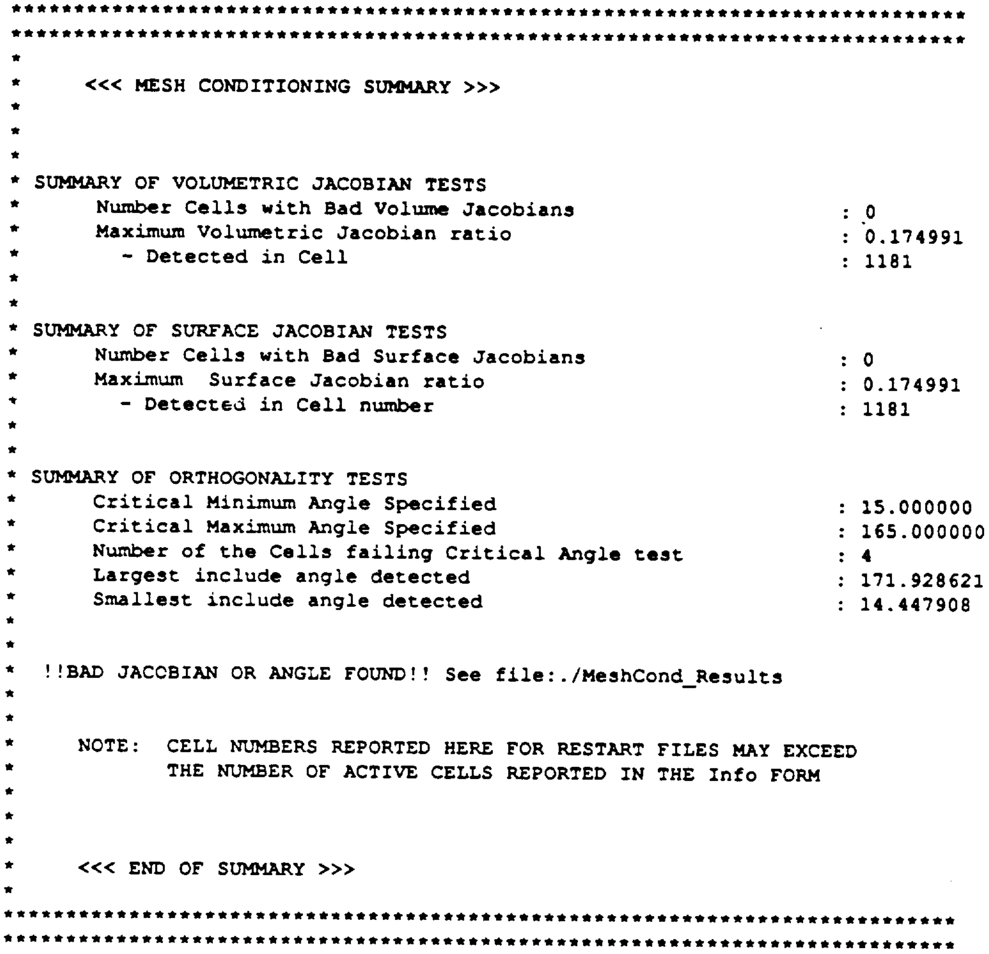


Figure 5.41: Computational mesh (near the airfoil) used by the JCODE to model inviscid M = 0.75 over a NACA 0012 geometry



```
**************************************************************************
**************************************************************************
*
*      <<< MESH CONDITIONING SUMMARY >>>
*
*
*
* SUMMARY OF VOLUMETRIC JACOBIAN TESTS
*       Number Cells with Bad Volume Jacobians                          : 0
*       Maximum Volumetric Jacobian ratio                               : 0.174991
*         - Detected in Cell                                            : 1181
*
*
* SUMMARY OF SURFACE JACOBIAN TESTS
*       Number Cells with Bad Surface Jacobians                         : 0
*       Maximum  Surface Jacobian ratio                                 : 0.174991
*         - Detected in Cell number                                     : 1181
*
*
* SUMMARY OF ORTHOGONALITY TESTS
*       Critical Minimum Angle Specified                                : 15.000000
*       Critical Maximum Angle Specified                                : 165.000000
*       Number of the Cells failing Critical Angle test                 : 4
*       Largest include angle detected                                  : 171.928621
*       Smallest include angle detected                                 : 14.447908
*
*
*  !!BAD JACCBIAN OR ANGLE FOUND!! See file:./MeshCond_Results
*
*
*     NOTE:  CELL NUMBERS REPORTED HERE FOR RESTART FILES MAY EXCEED
*            THE NUMBER OF ACTIVE CELLS REPORTED IN THE Info FORM
*
*
*
*     <<< END OF SUMMARY >>>
*
**************************************************************************
**************************************************************************
```

Figure 5.42: Preprocessor statistics for the NACA 0012 finite difference grid

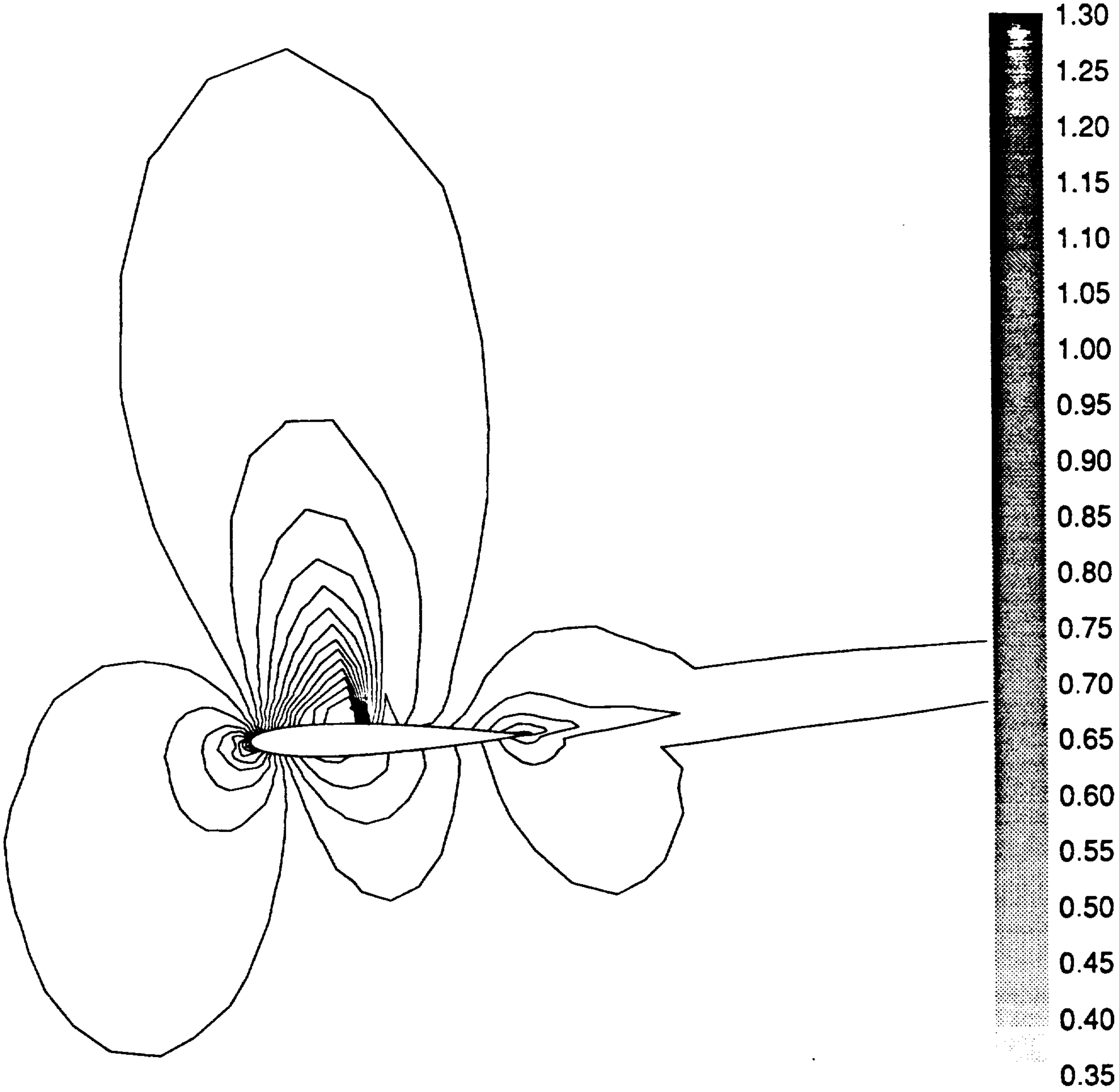


Figure 5.43: Mach number isosurfaces for NACA 0012 airfoil

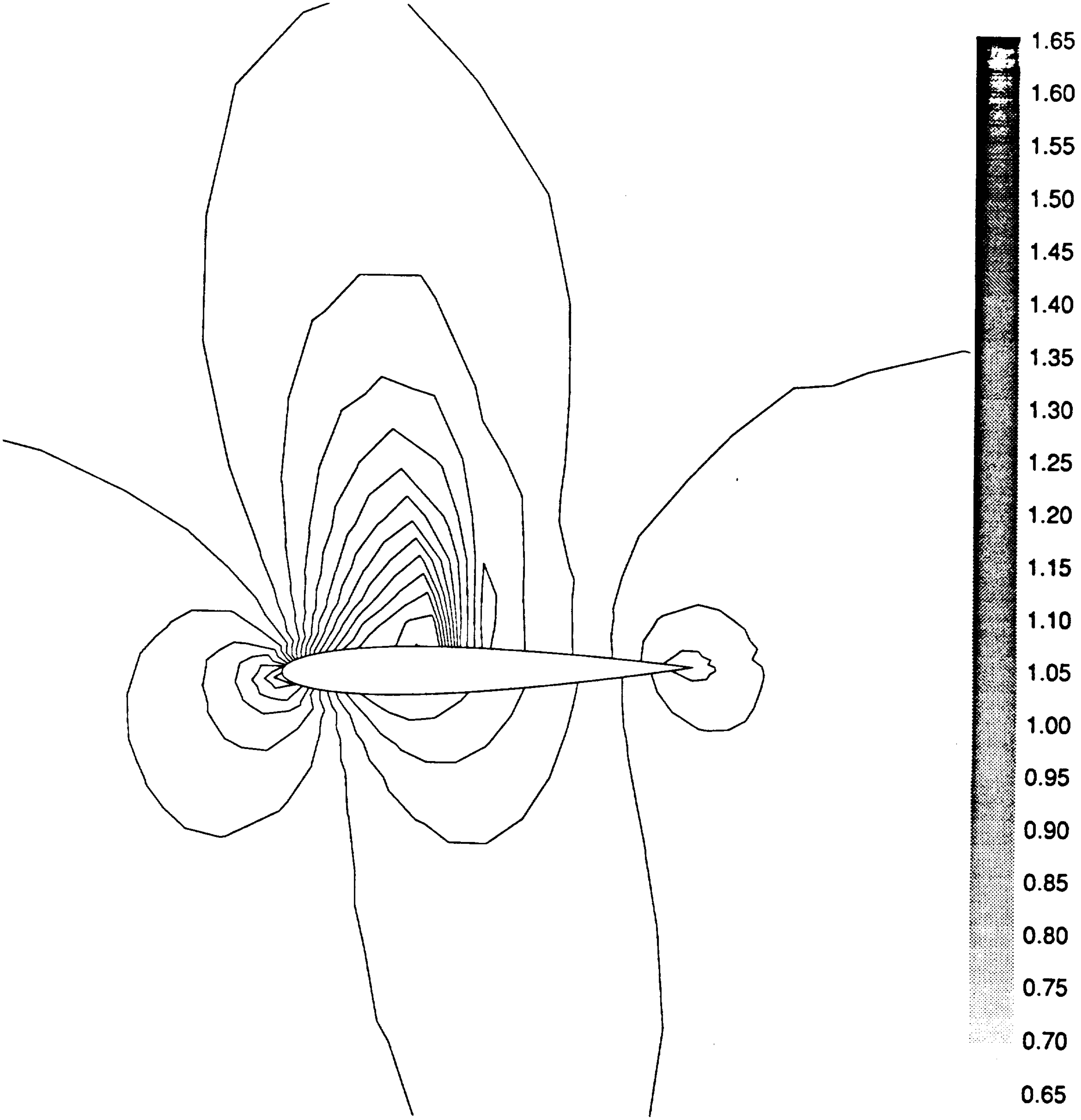


Figure 5.44: Pressure isosurfaces for NACA 0012 airfoil

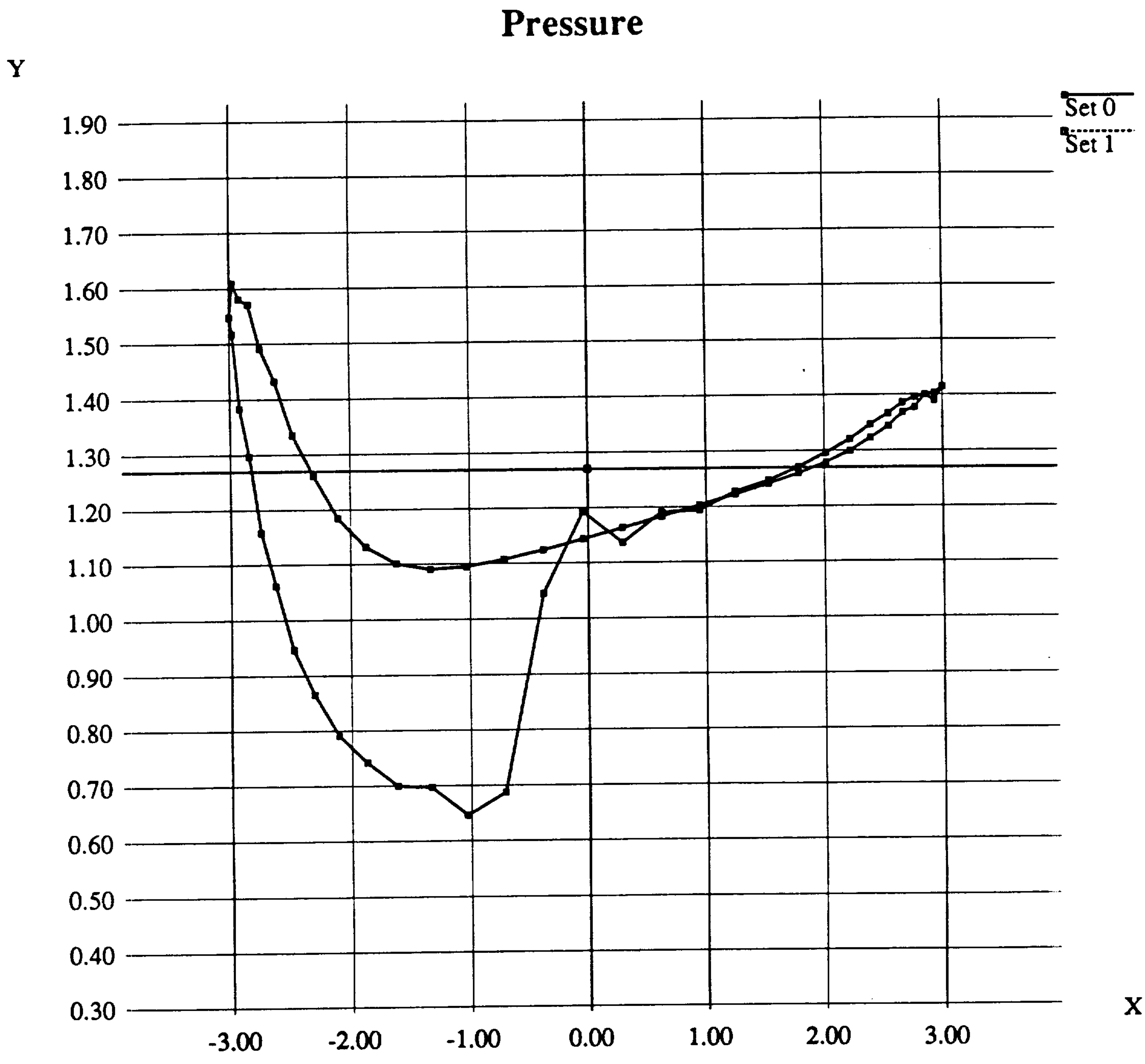


Figure 5.45: Pressure distribution along the wing surface predicted by the finite volume code JCODE.

5.5.2 Carter Flat Plate (P3/CFD - finite element solution)

The final compressible flow problem presented herein is a two-dimensional viscous flow past a flat plate. The geometry and boundary conditions used with the P3/CFD flow solver are shown in Figure 5.47. Summarizing the boundary conditions shown in this figure there is a supersonic inflow along the left face, openflow along the top face, outflow along the right face, a symmetry entrance length along part of the bottom face, and a solid wall plate along the remainder of the bottom of the domain.

On the inflow and openflow portions of the boundary the initial non-dimensional flow conditions are specified to be;

$$\begin{aligned} \text{Mach} &= 3.0 \\ \alpha &= 0.0 \text{ degrees} \\ \overline{p} &= 1.0 \\ \overline{u} &= 1.0 \\ \overline{v} &= 0.0 \\ \overline{w} &= 0.0 \\ \text{Re} &= 1000.0 \\ \text{Pr} &= 0.72 \\ \infty &= 390 \text{ R} \end{aligned}$$

The conditions on the solid wall plate were taken to be isothermal with a specified temperature of

$$\text{Twall} = 1092 \text{ R}$$

All quantities used in the computations are non-dimensionalized by the density and velocity at infinity.

Numerical Model

The geometry for this example, shown in Figure 5.47, has initially been discretized into 77 linear finite elements with 7 elements in the cross stream direction and 11 elements in the streamwise direction. This mesh was manually edited with a uniform refinement along the solid wall boundary, and then two global isotropic refinements were performed. The final mesh shown in Figure 5.48 contains 1712 elements and 1829 degrees of freedom.

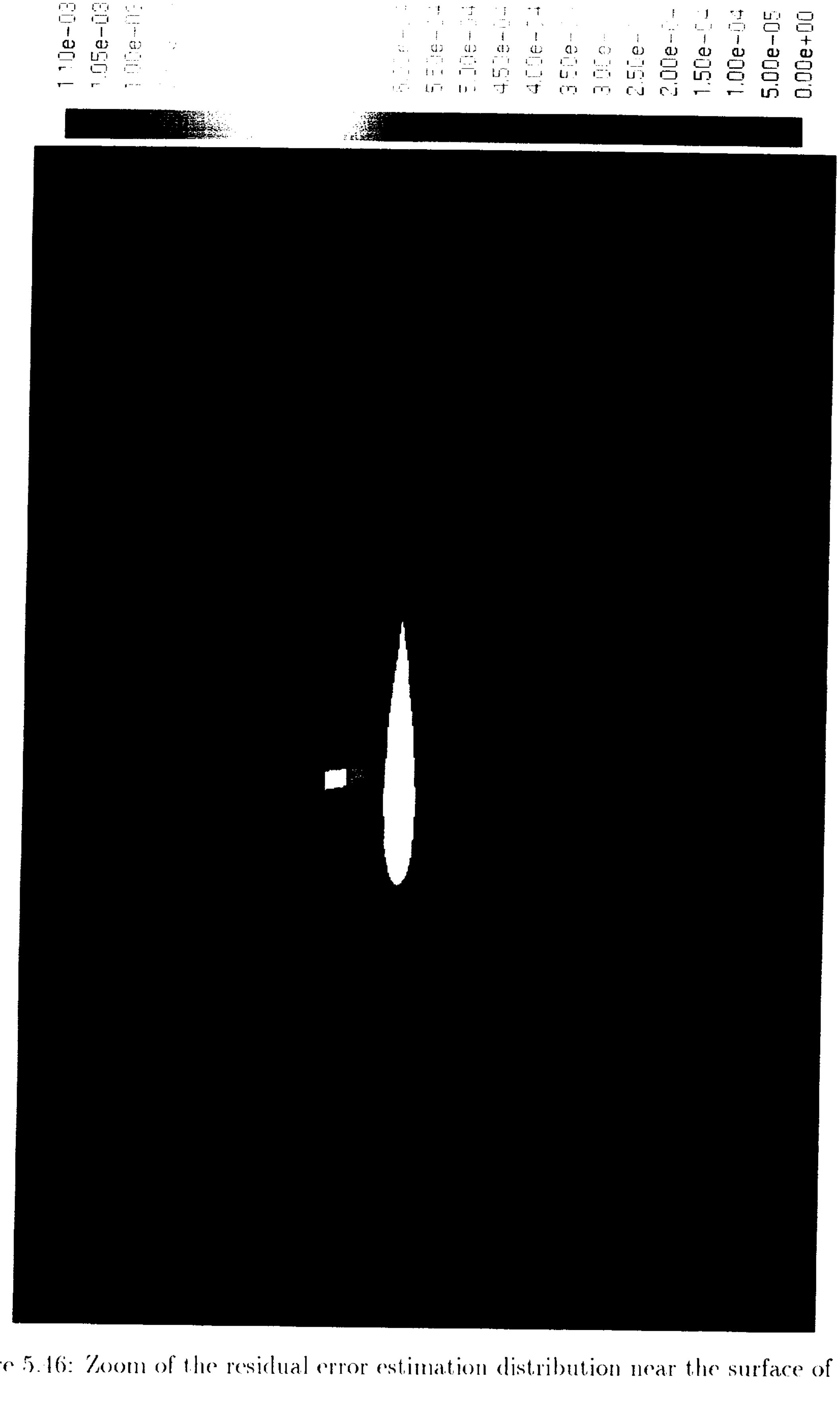


Figure 5.46: Zoom of the residual error estimation distribution near the surface of the wing

AUDITOR Analysis

The flow features of interest for this case include: the shock that develops at the tip of the plate and exits the right outflow boundary, the thickness of the boundary later region, and the distribution of the error in the numerical solution. To ensure that the numerical solution has been correctly read and placed in the database, a plot of the isosurfaces for the Mach number and the pressure along the plate are shown in Figures 5.49 and 5.50. These figures clearly show the shock emanating from the leading edge of the plate and exiting out the outflow boundary.

Activating the residual error estimator, the estimated global error in the solution is 21.7% with a maximum local error of 14.2%. Figure 5.51 shows the corresponding distribution of the residual error with the maximum local value cutoff at 5%. Based on the gray scale coloring in this figure, the maximum element errors are occurring at the leading edge of the plate and in the shock region. One can also observe some relatively large errors in the boundary layer region just down stream of thes singularity.

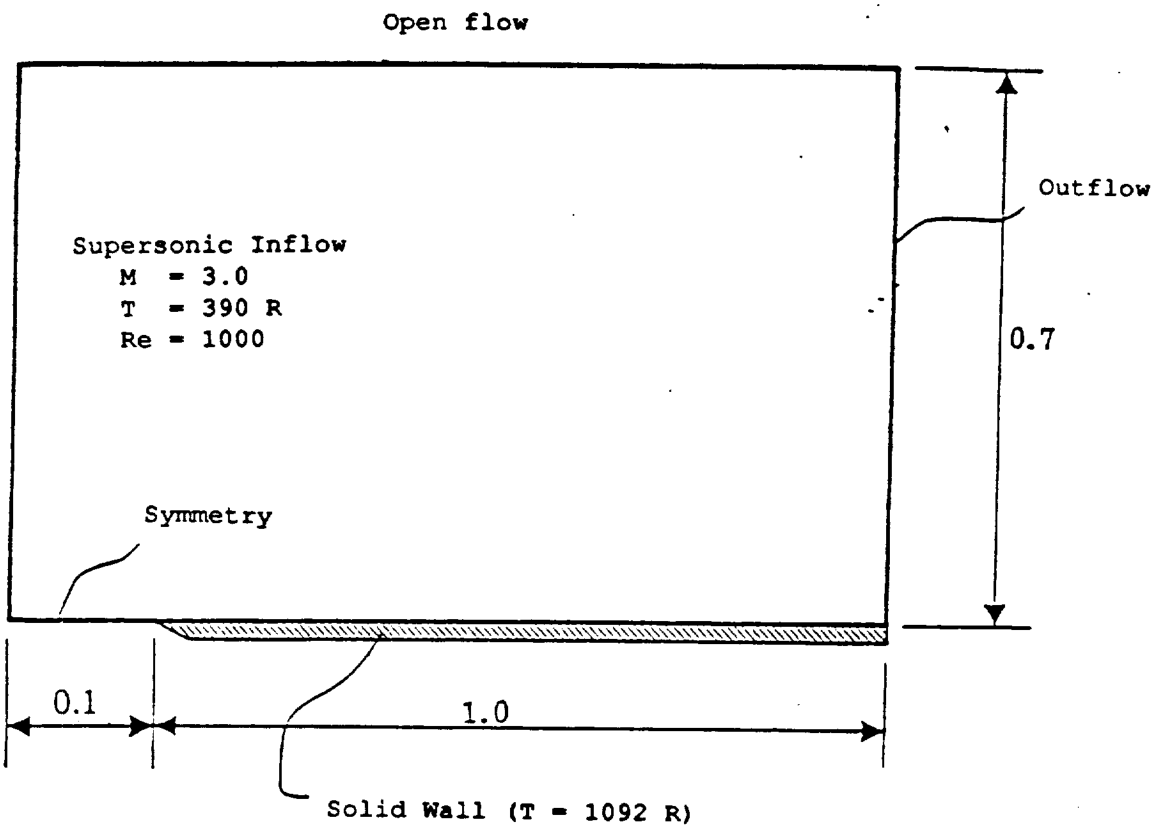


Figure 5.47: Carter's flat plate problem. Geometry and boundary conditions.

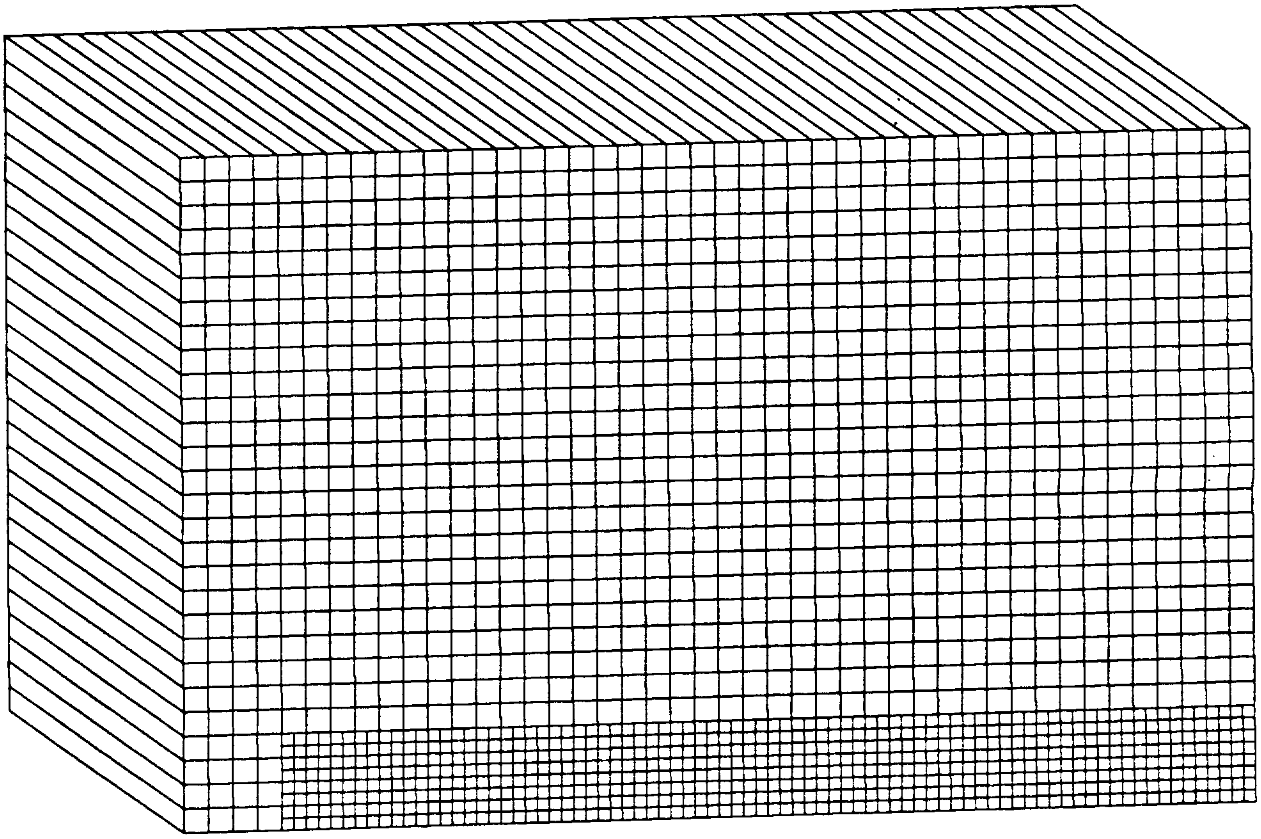


Figure 5.48: Computational mesh for the Carter flat plate problem, 1712 elements, 1829 degrees of freedom

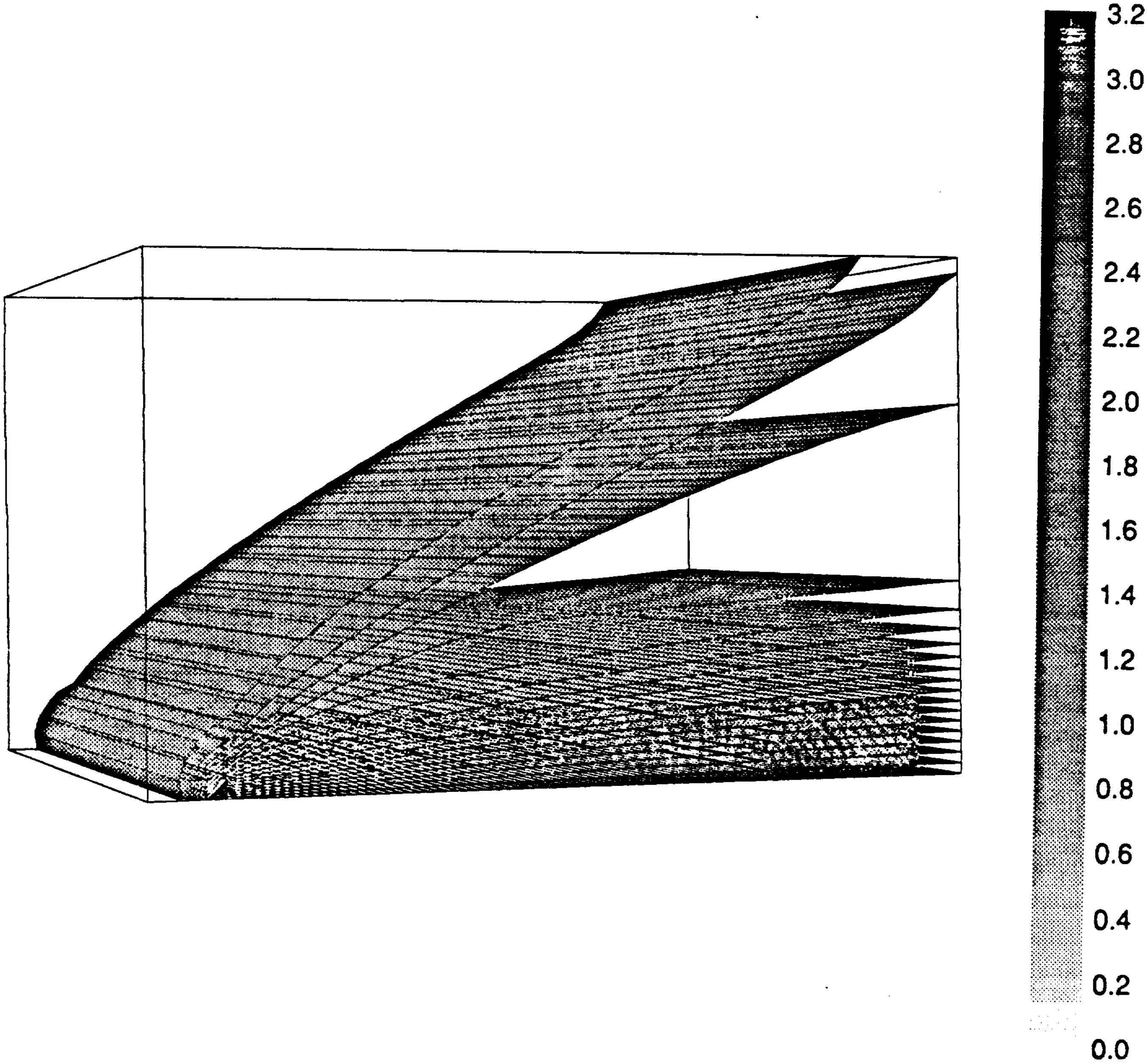


Figure 5.49: Mach isosurfaces for the Carter flat plate, Re = 1000

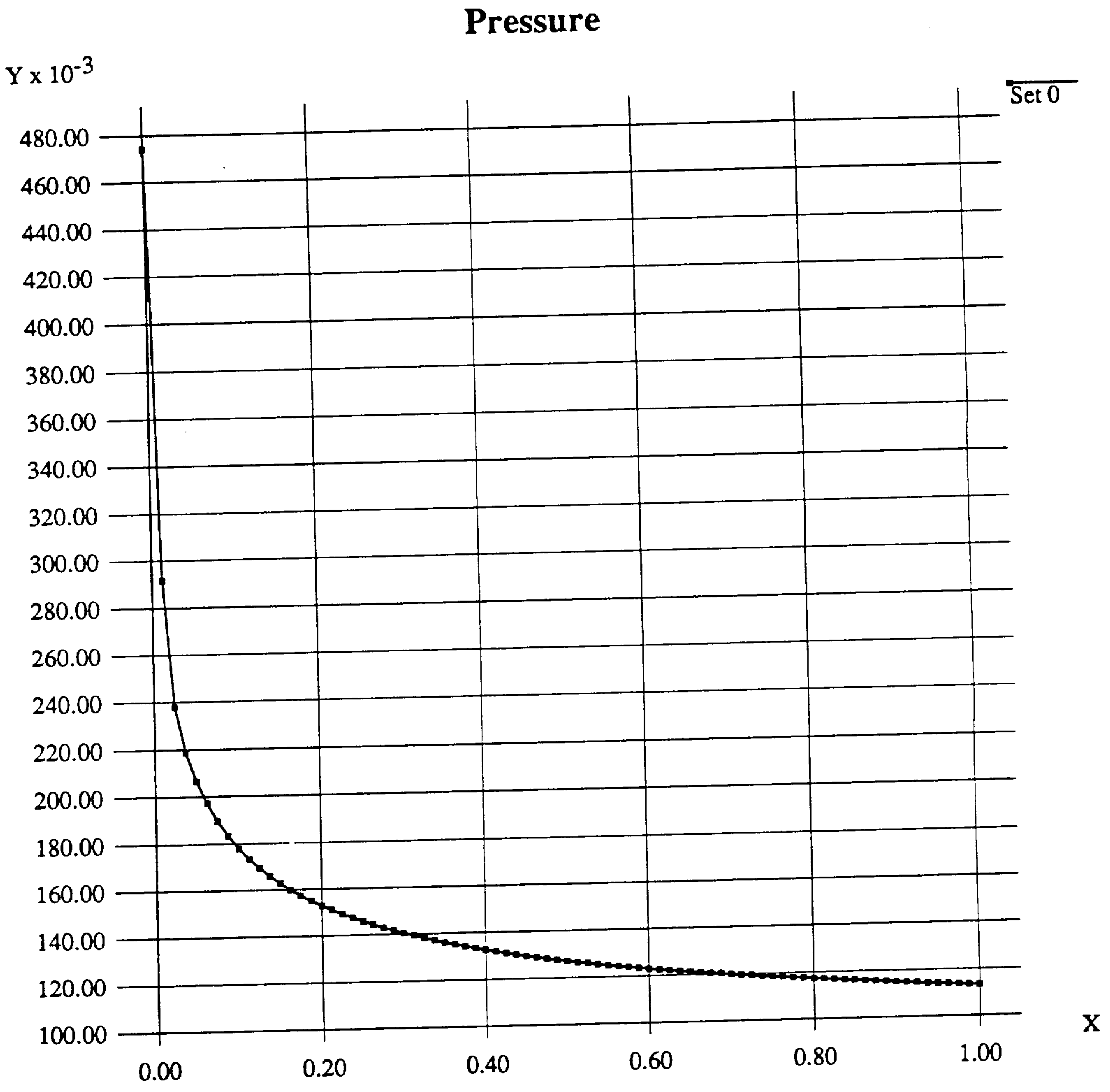


Figure 5.50: Pressure distribution along the surface of the Carter flat plate, Re = 1000

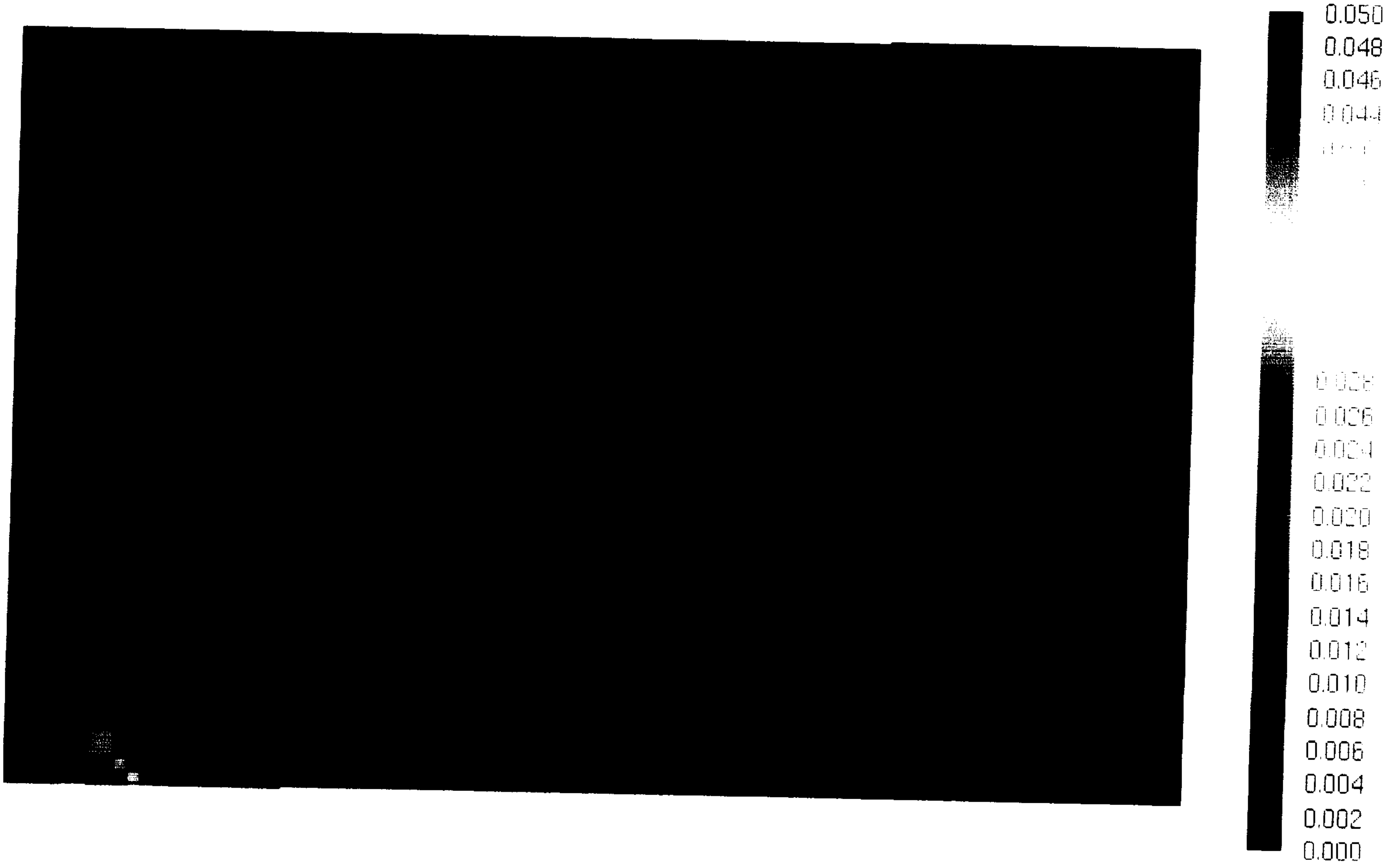


Figure 5.51: Residual error estimation distribution for the Carter flat plate with the maximum cell error limited to 5%

6 Future Directions

The Phase I feasibility study and the Phase II research and development effort have resulted in the development of a two- and three-dimensional pre- and postprocessing package called Auditor. The scope of this design includes a collection of new and unique features which we believe have advanced the state-of-the-art in computational mechanics. As with all research and development projects, several areas of research and development exist for which additional effort is needed to improve the robustness and utility of the current version of the Auditor code.

In particular, these areas include:

New Data Structure.

As indicated in Section 2, we have completed the design phase and much of the coding effort on a new object based data structure which is considerably more flexible than current object based hexahedral data structure. A reasonably focused development effort in this area could significantly expand the class of models (hexahedral, tetrahedral, prism, etc.) which Auditor accepts. This would obviously allow a much larger group of solutions to be read in and analyzed with Auditor.

Neutral File Reader.

As was also indicated in Section 2, we have completed the design phase and much of the coding effort on a neutral grid file reader. When completed, this reader will greatly simplify the conversion process required to read in finite volume and finite difference solutions as well as provide a means for reading in shell elements for elasticity problems.

Parallel Computations.

The current version of Auditor is designed to run in a workstation environment with some preliminary grouping and coloring of the mesh provided to aid in parallel computations. We believe that a moderate level of effort in this area could provide a workstation based version of Auditor which could operate with an efficiency of between 3 to 3.75 on a four processor workstation.

Batch Version.

Auditor was designed to operate interactively in a workstation environment. For large problems, the local computations of the element error and the solution enhancement can be computationally intensive. A batch version of Auditor which operates in a CRAY environment could provide significant increase in turn around time. Such an effort would require not only designing and implementing a batch interface for the code but also would require a reasonable effort toward vectorizing the package.

Error Estimation Scaling.

Auditor currently uses an estimate of the error based on residual calculation which is scaled by the energy norm of the solution for linear problems and an L2 norm of the solution for fluid flow problems. To provide the user a better insight on how various estimates of the error should be interpreted, further studies into error scaling are needed.

Directional Error Estimation.

Auditor presently provides a local estimate of the error on a cell-wise basis. The next obvious extension to this isotropic error estimate is to also develop an error estimate which has directional characteristics. Such a research effort could result in an error estimate that would be quite valuable for doing anisotropic mesh modifications.

Component Based Error Estimation.

For some problem classes, it may be desirable to allow the user to select a particular vector component for estimating the error in the solution. A rather short term effort would be required to modify the GUI and error estimation algorithms to allow the user to interactively select the fifth solution component, for example, on which to estimate the error.

As indicated above, some of these items are research issues while others are of a development nature. The incorporation of all of these features into Auditor would greatly enhance the utility and flexibility of the software package.

7 References

1. R. Carcaillet, G. Dulikravich, and S.R. Kennon, "Generation of Solution-Adaptive Computational Grids Using Optimization.", Comput. Methods Appl. Mech. Engng., **57**, pp. 279-295, (1986).

2. O. Jacquette, "A Mechanical Model for a New Grid Generation Method in Computational Fluid Dynamics.", Comput. Methods Appl. Mech. Engng., **66**, pp. 323-338, (1988).

3. O. Jacquette and J. Cabello, "A New Variational Method for the Generation of Two- and Three-Dimensional Adapted Grids in Computational Fluid Dynamics.", 7th International Conference for Finite Element Methods in Flow Problems, Huntsville, Alabama, April 3-7, 1989.

4. J.U. Brackbill, J.S. Saltzman,"Adaptive Zoning for Singular Problems in Two Dimensions.", *Journal of Computational Physics*, Vol. 46, pp. 342-368, 1982.

5. S. Kennon and D. Anderson, "Unstructured Grid Adaption for Non-Convex Domains," in Numerical Grid Generation in Computational Fluid Mechanics (Edited by S. Sengupta, J. Hauser, P.R. Eiseman, and J.F. Thompson), pp. 599-609. Pineridge Press. (1988).

6. L. Demkowicz and J.T. Oden, "On a Mesh Optimization Method Based on a Minimization of Interpolation Error", Int. G. Engng. Sci., **24**, pp. 55-68, (1986).

7. Demkowicz, L., Oden, J.T., and Strouboulis, T., "Adaptive Finite Element Methods for Flow Problems with Moving Boundaries. Part I. Variational Principles and *A Posteriori* Estimates," *Computer Methods in Applied Mechanics and Engineering*, Vol. 46, pp. 217-251, 1984.

8. Oden, J.T., Demkowicz, L., Strouboulis, T., and Devloo, P., "Adaptive Methods for Problems in Solid and Fluid Mechanics," *Accuracy Estimates and Adaptive Refinements in Finite Element Computations*, Edited by I. Babuska, O.C. Zienkiewicz, J. Gago, and E.R. de A. Oliveira, John Wiley and Sons, Ltd., Chichester, pp. 249-280, 1986.

9. Oden, J.T., Demkowicz, L., Rachowicz, W., and Westermann, T.A., "Toward a Universal $h--p$ Adaptive Finite Element Strategy, Part 2. *A Posteriori* Error Estimation," *Computer Methods in Applied Mechanics and Engineering*, Vol. 77, pp. 113-180, 1989.

10. Ainsworth, M. and Oden, J.T., "*A Unified Approach to A Posteriori Error Estimation Based on Element Residual Methods*, Numer. Math. (To appear), (1992).

11. Ainsworth, M. and Oden, J.T., "An Optimal Order Process for Calculating Self-Equilibrating Fluxes From $h-p$ Finite Element Approximations", (in preparation).

12. Kelly, D. W., "The Self-Equilibration of Residuals and Complimentary Error Estimates in the Finite Element Method," *Int. J. Num Meth. Eng.*, Vol. 20, pp. 1491-1506, 1984.

13. Babuska, I. and Miller, A.D., "A Feedback Finite Element Method With *A Posteriori* Estimation: Part 1," *Comp. Meth. Appl. Mech. Eng.*, Vol 61, pp. 1-40, 1987.

14. Strang, G. and Fix, G.J., "An Analysis of the Finite Element Method", Prentice-Hall, 1972.

15. Ainsworth, M. and Craig, A.W., *A Posteriori error estimators in the finite element method*, Numer. Math., pp. 429-463, **60**, (1991).

16. Zienkiewicz, O.C and Zhu, J.Z., "*A simple error estimator and adaptive procedure for practical engineering analysis*, Int. J. Num. Meth. Eng., **24**, pp. 337-357, (1987).

17. Babuska, I. and Miller, A., "*The postprocessing approach in the finite element method. Part I: Calculation of displacements, stresses and other higher derivatives of the displacements*, Int. J. Num. Meth. Eng., **20**, pp. 1085-1109, (1984).

18. Babuska, I. and Miller, A., *The postprocessing approach in the finite element method. Part II: The calculation of stress intensity factors*, Int. J. Num. Meth. Eng., **20**, pp. 1111- 1129, (1984).

19. Babuska, I. and Miller, A., *The postprocessing approach in the finite element method. Part III: A posteriori error estimates and adaptive mesh selection*, Int. J. Num. Meth. Eng., **20**, pp. 2311- 2324, (1984).

20. Krizek, M. and Neitaanmaki, P. *On superconvergence techniques*, Acta Applicandae Mathematicae, **9**, pp. 175-198, (1987).

21. Noor, A.K. and Babuska, I., *Quality assessment and control of finite element solutions*, Finite Elements and Design, **3**, pp. 1-26, (1987).

22. Handscomb, D.C., *Recovery of Divergence Free Vector Fields*, Oxford University Computing Laboratory, Unpublished, (1992).

23. Ainsworth, M., Oden, J.T., and Wu, W., *A Posteriori Error Estimators for h-p Finite Element Approximations in Linear Elastostatics*, (To appear), (1992).

24. Ainsworth, M. and Oden, J.T., *A procedure for a posteriori error estimation for h-p finite element methods*, Comp. Meth. Appl. Mech. Eng., (To appear), (1992).

25. Ainsworth, M. and Oden, J.T., *A Posteriori Error Estimators for Second Order Elliptic Systems: Part 2. An Optimal Order Process for Calculating Self Equilibrating Fluxes*, Computers and Mathematics, (To appear), (1992).

26. Oden, J.T., Demkowicz, L., Rachowicz, W., and Westermann, T.A., *A posteriori error analysis in finite elements: the element residual method for symmetrizable problems with applications to compressible Euler and Navier Stokes equations*, Comp. Meth. Appl. Mech. Eng., **82**, pp. 183-203, (1990).

27. Ainsworth, M. and Oden, J.T., *A Posteriori Error Estimators for Second Order Elliptic Systems: Part 1. Theoretical Foundations and A Posteriori Error Analysis*, Computers and Mathematics, (To appear), (1992).

28. Strang, G. and Fix, G.J., **An Analysis of the Finite Element Method**, Prentice-Hall, 1972.

29. Barlow, J., "Optimal Stress Location in the Finite Element Analysis," *Int. J. Num. Meth. Eng.*, **10**, pp. 243-251, 1976.

30. Wheeler, M.F. and Whiteman, J.R., "Superconvergent Recovery of Gradients on Subdomains from Piecewise Linear Finite Element Approximations," *Numerical Methods for Partial Differential Equations*, **3**, pp. 65-82, 1987.

31. Andreer, A.D. and Lazarov, R., "Superconvergence of Gradient for Quadratic Triangular Finite Elements", *Numerical Methods for Partial Differential Equations*, **4**, pp. 15-32, 1988.

32. Lesaint, P. and Zlamal, M., "Superconvergence of the Gradient of Finite Element Solutions," *RAIRO Analyse Numerique*, **13**, pp. 139-166, 1979.

33. Brauchli, H.J. and Oden, J.T., "Conjugate Approximations in Finite Element Analysis," *Quarterly of Applied Mathematics*, 1971.

34. Hinton, E. and Campbell, J.S., "Local and Global Smoothing of Discontinuous Finite Element Functions Using a Least Squares Method," *Int. J. Num. Meth. Eng.*, **8**, pp. 461-480, 1974.

35. Rachowicz, W., and Oden, J.T., "On the Accuracy and Convergence of Conjugate Flux Approximations," *Numerical Methods for Partial Differential Equations*, Vol. 5, pp. 143-156, 1989.

36. Bramble, J.H. and Schatz, A.H., "Higher Order Accuracy by Averaging in the Finite Element Method," *Math. Comp.*, **31**, pp., 94-111, 1977.

37. Ainsworth, M., Zhu, J.Z., Craig, A.W., and Zienkiewicz, O.C., "Analysis of the Zienkiewicz-Zhu Error Estimator," *Int. J. Num. Meth. Eng.*, **28**, pp. 2161-2174, 1989.

38. Rachowicz, W., "An Evaluation and Comparison of Postprocessing Methods for Finite Element Solution of Elliptic Boundary Value Problems," TICOM Report 87-11, 1987.

39. Babuska, I. and Miller, A.D., "The Postprocessing Approach in the Finite Element Method -Part 1: Calculation of Displacements, Stresses, and Other Higher Derivatives of Displacements," *Int. J. Num. Meth. Eng.*, **20**, pp. 1085-1109, 1984.

40. Babuska, I. and Miller, A.D., "The Postprocessing Approach in the Finite Element Method -Part 2: The Calculation of Stress Intensity Factors," *Int. J. Num. Meth. Eng.*, **20**, pp. 1111-1129, 1984.

41. Babuska, I. and Miller, A.D., "The Postprocessing Approach in the Finite Element Method -Part 3: *A Posteriori* Error Estimates and Adaptive Mesh Selection," *Int. J. Num. Meth. Eng.*, **20**, pp. 2311-2324, 1984.

42. Stern, M., Becker, E.B. and Dunham, R.S., "A Contour Integral Computation of Mixed Mode Stress Intensity Factors," *Int. J. Fracture*, **12**, pp. 359-368, 1976.

43. Wheeler, J.A., "Simulation of Heat Transfer from a Warm Pipeline Buried in Permafrost," presented at the 74th Meeting of American Institute of Chemical Engineers, New Orleans, March 1973.

44. Douglas, J., Dupont, T.F., and Wheeler, M., "A Galerkin Procedure for Approximating the Flux on the Boundary for Elliptic and Parabolic Boundary Value Problems," *RAIRO, Analyse Numerique*, **8**, pp. 47-59, 1974.

45. M. Zlamal, "Superconvergence and Reduced Integration in the Finite Element Method.", Math. Comp., **32**, pp. 663-685, (1978).

NASA
National Aeronautics and
Space Administration

Report Documentation Page

1. Report No.
TR-93-06

2. Government Accession No.

3. Recipient's Catalog No.

4. Title and Subtitle
Pre- and Postprocessing Techniques for Determining Goodness of Computational Meshes

5. Report Date
April 27, 1993

6. Performing Organization Code

7. Author(s)

J. Tinsley Oden, T. Westermann, and J. M. Bass

8. Performing Organization Report No.

10. Work Unit No.
SBIR 88-1-II 02-01

9. Performing Organization Name and Address

Computational Mechanics
7701 North Lamar, Suite 200
Austin, Texas 78752

11. Contract or Grant No.
NAS8-38478

12. Sponsoring Agency Name and Address

NASA
Washington, DC 20546-0001
MSFC, AL

13. Type of Report and Period Covered
Final 9/11/90 - 4/23/93

14. Sponsoring Agency Code

15. Supplementary Notes

None

16. Abstract

Research in error estimation, mesh conditioning, and solution enhancement for finite element, finite difference, and finite volume methods has been incorporated into AUDITOR, a modern, user-friendly code, which operates on 2D and 3D unstructured neutral files to improve the accuracy and reliability of computational results.

Residual error estimation capabilities provide local and global estimates of solution error in the energy norm. Higher order results for derived quantities may be extracted from initial solutions. Within the X-MOTIF graphical user interface, extensive visualization capabilities support critical evaluation of results in linear elasticity, steady state heat transfer, and both compressible and incompressible fluid dynamics.

17. Key Words (Suggested by Author(s))

Error estimation, solution enhancement, mesh preprocessing, finite - element, volume, and difference

18. Distribution Statement

Pursuant to Contract Attachment J-2, D. and Rights in Data Clause SBIR Program (April 1985)

9. Security Classif. (of this report)	20. Security Classif. (of this page)	21. No. of pages	22. Price
Unclassified	Unclassified	233	n/a

Final Report

on

PRE– AND POSTPROCESSING TECHNIQUES FOR DETERMINING "GOODNESS" OF COMPUTATIONAL MESHES

Contract #NAS8-38478

National Aeronautics and Space Administration
Marshall Space Flight Center

TR-93-06
April 1993

COMCO

Computational Mechanics Company, Inc.
7701 North Lamar, Suite 200
Austin, Texas 78752
(512) 467-0618

PROJECT SUMMARY

PURPOSE:

To improve the quality and reliability of numerical simulations in computational mechanics.

AREAS OF RESEARCH:

Error estimation, solution enhancement, and mesh conditioning for finite element, finite volume, and finite difference meshes and solutions for problems in linear elasticity, steady state heat conduction, viscous incompressible flow, viscous and inviscid compressible flow.

RESEARCH RESULT:

An operational software package, AUDITOR, incorporating the following advanced features:

- Accepts unstructured 2D and 3D neutral files for finite element, finite volume, and finite difference meshes of arbitrary spectral order;
- Incorporates an object-based data structure;
- Pre-processes meshes for conditioning characteristics;
- Computes local and global solution error with advanced residual error estimation techniques;
- Post-processes initial solutions to derive enhanced, higher order results;
- Provides advanced 3D visualization;
- Operates in a modern, X-MOTIF graphical user interface.

POTENTIAL APPLICATIONS:

The AUDITOR code may be operated as a standalone adjunct to many commonly used simulation codes, or its features may be integrated into such codes, thereby enhancing their reliability and robustness.

Contents

List of Figures

List of Tables

1 Introduction

This report is the final report for the Phase II project entitled "Pre- and Postprocessing Techniques for Determining Goodness of Computational Meshes". This project has focused on the development of a number of new ideas and methodologies which could significantly improve the quality and reliability of numerical simulations in computational mechanics. The methodologies are based on the concepts of error estimation, solution enhancement, and mesh optimization each of which plays a integral part in the success of the modeling effort and reliability of the numerical results. The ultimate goal of this effort is to provide a standalone pre- and postprocessing package which is capable of accepting arbitrary unstructured meshes and numerical solutions from various class of numerical solver techniques and provide graphical and tabulated feedback on the quality of the mesh and reliability of the solution.

With this rather general set of objectives in mind, a Phase I research and development effort was undertaken aimed at both a-priori and a-posteriori improvement of the quality of numerical solutions for a large class of linear and nonlinear computational mechanics problems. In particular, this effort focused on error estimation, solution enhancement, and mesh conditioning for finite element, finite volume, and finite difference computational meshes and solutions for linear elasticity, steady state heat conduction, viscous incompressible flow, and inviscid and viscous compressible flows. The specific objectives of the effort were as follows:

- The development of two-dimensional and three-dimensional preprocessing techniques which provide the user with an a-priori indication of the quality of the initial computational mesh.
- The development of two-dimensional and three-dimensional a-posteriori error estimation techniques for accessing the cell-wise errors in user supplied numerical solutions. These estimates are based on residual error estimation techniques which provide both a local and global estimate of the error in the numerical solution.
- Post-processing methods were to be developed for two-dimensional and three-dimensional simulations which provide for higher order solution extraction.
- Graphics and visualization capabilities were to be developed to provide the user with a graphical interface to view the computational domain, the solution components, extracted enhanced solution data, and distributions of the solution error.

This report contains detailed descriptions of the work done toward meeting all of these objectives during the course of the Phase II effort. In particular, a summary of the research and development progress is provided in Section 2. This is followed in Section 3 by an overview of the theory, algorithms, and methodologies studied in the course of this Phase II effort, many of which have been incorporated into the *AUDITOR* pre- and postprocessing package. In Section 4, an overview of the functionalities and features which comprise the

AUDITOR package are presented. This is followed in Section 5 by a summary of the results of several benchmark problems that have been solved to validate various components of the code. In particular, representative results for two-dimensional and three-dimensional heat conduction, linear elasticity, compressible and incompressible flows are presented here. Finally, Section 6 concludes this report with some observations about potential areas for continued research and development.

2 Summary of the Phase II Effort

The principal objectives of the Phase II research and development effort have been outlined in Section 1. The ultimate goal of this project, however, has been to synthesize these features and functionalities into a fully operational tool for the two- and three-dimensional pre- and post-postprocessing of user supplied computational meshes and numerical solutions. The culmination of this effort has resulted in what we believe is a unique software package that contains features and analysis capabilities that are collectively not available in any other commercial or private analysis package. In particular, we believe that the residual error estimation capabilities, the ability to accept unstructured meshes and solutions from finite element, finite volume, and finite difference solvers, the scope of the class of problems accepted (linear elasticity, steady state heat conduction, compressible and incompressible fluid dynamics), and the ability to audit hp-adapted meshes, represents a significant advance in the area of computational mechanics.

The technical accomplishments that have been reached during the Phase II effort are summarized as follows;

- The development of residual error estimation capabilities which incorporate the following features or functionalities
  - are applicable to finite element, finite volume, or finite difference numerical solutions
  - provide local (cell-wise) and global estimates of the solution error in the energy norm
  - provide accurate error estimates of solution error for unstructured meshes which may be composed of cells with arbitrary spectral order
  - do not require the specification of boundary conditions; the appropriate boundary conditions are assumed to be satisfied by the approximate solutions.
  - solve a sequence of local problems to obtain estimates of the error and thus avoids the assembly and solution of a large system of equations
  - provide an interface to the graphics package to view the results of the error estimation process
- The development of mesh postprocessing capabilities which incorporate the following features:
  - are applicable to finite element, finite volume, or finite difference numerical solutions
  - extract enhanced higher order results for derived quantities based on the initial solution

  - provide and interface to the graphics package to view the results of the enhancement process

- The development of mesh preprocessing capabilities for performing an initial mesh preconditioning analysis. In particular, the efforts in this area have resulted in the following:

  - a mesh preconditioning module which checks the smoothness and orthogonality of the initial mesh
  - a module which allows the user to interactively specify criteria on which to test the mesh characteristics
  - a diagnostic feed back both in a tabular form and graphically though the graphics/visualization package.

- The development of motif graphical user interface which is mouse driven and is compatible with X_window platform.

- The development of various components of a three-dimensional graphics package for viewing unstructured meshes and solutions. Among the features which comprise this package are;

  - isosurface plotting for both volumes and surfaces
  - planar slicing
  - velocity vectors
  - xy-plotting capabilities
  - local error estimate plotting capabilities
  - mouse driven zoom, pan, rotate, and tumble features
  - postscript hard copy capabilities

- The development of gridfile interface with the following options;

  - reads a gridfile in a predefined COMCO format (see the *AUDITOR* user manual for details)
  - reads a neutral file generated from P3/CFD or PHLEX

In addition to these milestones significant progress has also been made in the following areas:

- A generic neutral file reader has been developed which accepts unstructured two and three-dimensional computational meshes in a given format and directly translates this data into cell wise data that is loaded into the *AUDITOR* database. The scope of this package includes:

  - a capability for accepting finite element, finite volume, or finite difference meshes and solutions in various formats
  - a capablility for accepting various types of solution data including solution derivatives at arbitrary specified locations
  - a capability for converting various element types and associated data formats

- The development of an extended object based data structure which includes:
  - dynamic memory management
  - objects structures and methods
  - stacks, heaps, and queues
  - manipulates multiple element types
    * hexahedral elements
    * tetrahedral elements
    * prism elements
    * shell elements

The final code incorporates some features that are common to other software packages that are under development at COMCO. These include the current data structure on which *AUDITOR* is built, and the graphics and postprocessing capabilities. In addition, initial support for Phase III productization of some of this technology in the area of computational fluid dynamics has been received by a partner company and could result in a commercial release in a new residual error estimation feature for P3/CFD within the next six to twelve months.

3 Technical Approach

This section describes the technical aspects of the methods used in *AUDITOR* to perform the various tasks. Also included in this section are summaries of the research studies performed during the project in developing these methods.

3.1 Mesh Preprocessing

The mesh conditioning module analyzes the mesh, quantifying its various attributes and searching for inconsistencies. This analysis is usually done by evaluating special functionals designed to quantify various desirable features of a mesh, such as smoothness and orthogonality. The definition of these functionals varies from author to author, but all of them are invariably dependent upon the geometric features of the mesh.

For example, Carcaillet, et. al. [1] use functionals based upon vectors oriented along cell edges. Smoothness is defined according to

$$F_{\text{smooth}} = \sum_{i=1}^{\#\text{nodes}} \sum_{j=1}^{\#\text{edges(i)}} ||\boldsymbol{r}_{ij}||^2$$

where i is the set of corner nodes, j is the set of edges connected to node i and $||\boldsymbol{r}_{ij}||$ is the length of a particular edge. Orthogonality is measured according to the functional

$$F_{\text{orth}} = \sum_{i=1}^{\#\text{nodes}} \sum_{j=1}^{\#\text{faces(i)}} (\boldsymbol{r}_{ij1} \cdot \boldsymbol{r}_{ij2})^2$$

where i is the set of corner nodes, j is the set of faces connected to node i and $\boldsymbol{r}_{ij1}$ and $\boldsymbol{r}_{ij2}$ are the two edges on face j connected to node i.

Other authors, such as Jacquotte [2,3], have defined functionals based on cell deformation ideas. Specifically, these functionals use the invariants of the left Cauchy-Green tensor of the deformation of a reference cell into an actual cell. In two-dimensions, this functional takes on the form

$$F = C_1(I_1 - 2J) + C_2(J - 1)^2$$

and in three-dimensions

$$F = C_1(I_1 + I_2 - 6J) + C_2(J - 1)^2$$

where I_1 and I_2 are the invariants of the left Cauchy-Green tensor $\boldsymbol{C}$ of the transformation from a reference cell into a physical cell $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$. J is the Jacobian of the transformation and C_1 and C_2 are constants used to weight the two terms. These quantities can be written

$$\begin{aligned} I_1 &= \operatorname{tr} \boldsymbol{C} \\ I_2 &= \operatorname{tr} \operatorname{Cof} \boldsymbol{C} \\ J &= \det \boldsymbol{\nabla x} \end{aligned}$$

In these functionals, the first term measures the "conformality" of the deformation while the second term is a volume control term. Together, these two terms measure the smoothness and orthogonality of the mesh.

The most common technique for measuring smoothness and orthogonality, however, was introduced by Brackbill and Saltzman [4]. This method uses the transformation from a reference (ξ,η) to physical space (x,y) to define functionals for smoothness

$$F_{\text{smooth}} = \int_{\hat{\Omega}} [(\boldsymbol{\nabla}\xi \cdot \boldsymbol{\nabla}\xi) + (\boldsymbol{\nabla}\eta \cdot \boldsymbol{\nabla}\eta)] d\xi d\eta$$

and orthogonality

$$F_{\text{orth}} = \int_{\hat{\Omega}} [(\boldsymbol{\nabla}\xi \cdot \boldsymbol{\nabla}\eta)^2] d\xi d\eta$$

Most authors, such as Kennon [5] and Demkowicz [6] however, write these functionals in terms of the transformation $\boldsymbol{x} = \boldsymbol{x}(\boldsymbol{\xi})$

$$F_{\text{smooth}} = \int_{\hat{\Omega}} \left[\left(\frac{\partial x^2}{\partial \eta} \right) + \left(\frac{\partial y^2}{\partial \eta} \right) + \left(\frac{\partial x^2}{\partial \xi} \right) + \left(\frac{\partial y^2}{\partial \xi} \right) \right] \frac{1}{J} d\xi d\eta$$

and

$$F_{\text{orth}} = \int_{\hat{\Omega}} \left(\frac{\partial x}{\partial \xi}\frac{\partial y}{\partial \xi} + \frac{\partial x}{\partial \eta}\frac{\partial y}{\partial \eta} \right)^2 \frac{1}{J} d\xi d\eta$$

As the work of these authors indicates, the condition of the mesh can be measured by examining the transformations from the reference cell to the physical cells. In particular, this can be done by evaluating the jacobians of the transformation from the master to physical element for each element in the mesh. The values of the jacobians and their determinants can be used to indicate several different measures of mesh quality. For example, a negative determinant indicates an element which is "inside out" or at least partially folded over on itself. Wide variations in the values of the determinant in an element imply a highly distorted element with very small and/or large angles which often result in a poor local approximation. Furthermore, a non-constant determinant usually results in a certain amount of error when quadrature rules are used to integrate over the element. This error may become significant the more the determinant deviates from a constant especially if it becomes very small.

As a result, mesh analysis based upon the element jacobians would identify highly distorted elements and elements with bad (negative) determinants, and then allow the user to view the offending elements. Furthermore, the analysis would evaluate statistical measures of the variations of the jacobians in order to provide an overall indicator of the mesh quality.

3.2 Error Estimation Techniques and Research

A significant portion of the work performed on this project was spent researching error estimation techniques which would be suitable for the wide range of problems under consideration (linear convective heat transfer through compressible viscous flow). This section summarizes these efforts following a brief introduction on the purposes of error estimation.

3.2.1 Why Bother to Estimate Errors?

The first, foremost, and obvious reason an analyst would want to estimate errors in an approximate solution is reliability. Simply put, if engineering design or analysis decisions are to be made based upon data produced by an approximate solution technique, this technique must have some way of providing some assurance that the calculated data is accurate. Unfortunately, most engineering analyses rely primarily upon the expertise of the user. While there is no substitute for experience, a user simply does not have enough information to make an informed judgment. Traditionally, when examining a solution, a user will look for inconsistencies in the solution or changes in the solution from one mesh to the next. For example, in an elasticity problem, a user might look for discontinuities in the computed stresses or changes in the strain energy of the computed solution from one mesh to the next. (Some analysis programs refer to this change in the energy of the solution as the "error in energy.") While these techniques indicate when error is present, the converse is not true. A computed solution may have continuous derivatives and may not change significantly from mesh to mesh but still be inaccurate. Consequently, *these techniques used to analyze a computed solution have no direct relationship with the actual error*; they merely are features that the error may or may not exhibit.

In reality, when attempting to estimate a solution's reliability, the user wishes to know the difference between the computed solution and the solution which satisfies the governing differential equation and boundary conditions exactly. Furthermore, the user may not be interested in the solution but in some combination of its derivatives (such as stresses, heat fluxes, or skin friction forces) and may be interested in the values of these quantities only in certain places. Suppose the analytic solution were known. Figure 3.1 shows the computed solution, the error in the computed solution, and the errors in the solution's x- and y-derivatives for a mesh of 18 quadratic elements where the analytic solution is a simple bi-cubic polynomial. A reliability analysis would want to know how good the solution is in some region of the mesh. As Fig. 3.1 clearly demonstrates, even with knowledge of the analytic error, a reliability analysis can still be difficult to perform.

For this reason and others, error estimates are generally calculated in terms of a norm of the estimated error which in turn is usually "normalized" by the same norm of the solution. For example, if the L_2 norm is being used, the error norm for a particular computational cell, Ω_k would be:

$$\frac{\left[\int_{\Omega_k} (u - u_h)^2 \, d\Omega\right]^{\frac{1}{2}}}{\left[\int_{\Omega_k} (u)^2 \, d\Omega\right]^{\frac{1}{2}}}$$

where u and u_h represent the analytic and computed solutions, respectively. In this form, the error norm represents an "average" error over the computational cell as a percentage of the solution (i.e., an L_2 error norm of 0.05 would imply that the computed solution is on the average about 5 percent different from the analytic solution).

In a typical error estimation analysis, the above error function, $u - u_h$, is replaced by an estimate, e_h, and the estimate is normalized by the norm of the computed solution. Additionally, since the user is often interested in the derivatives of the computed solution, the H_1 norm or the "energy" norm is used instead of the L_2 norm.

Notice that in order to compute an error estimate norm, the estimated error must first be calculated. Under certain circumstances, this function estimating the error can be added to the computed solution resulting in a new solution with greater accuracy.

In addition to determining the reliability of a computed solution, error estimates provide information to the user which can be used to change the mesh to improve the solution. This idea is based upon the premise that replacing a cell that has a large error with a number of smaller cells (or a cell with a higher spectral order) will reduce the error in that region of the mesh. For diffusion dominated problems, this premise is rarely violated. For problems with significant convective phenomena, however, the errors in elements which are "upstream" of the element must also be considered.

3.2.2 The Dual Norm and Residual Error Estimates

Perhaps the most natural functional setting for the finite element method is to assume that solution space U is a subspace of H^1, whereas space V is a topological dual of U, $V = U'$. This requires that the residual is to be measured using the dual space norm

$$\|r_h\|_{U'} = \sup_{\substack{v \in U \\ v \neq 0}} \frac{\langle r_h, v\rangle}{\|v\|} \tag{1}$$

This idea has led to the element residual method which will be presented in this and the following sections. Furthermore, this technique was carefully examined in a number of papers

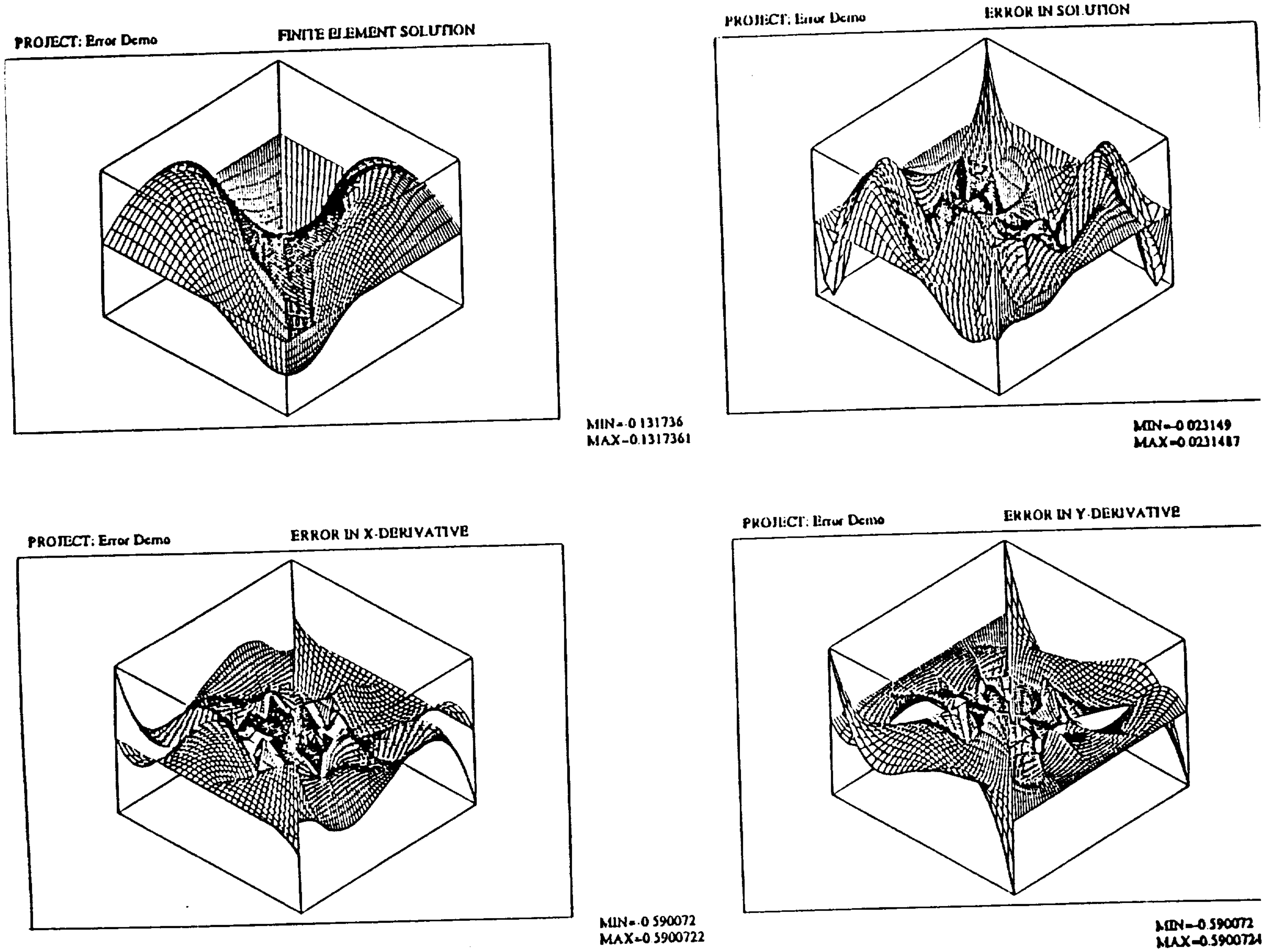


Figure 3.1: Computed solution, error in computed solution, and error in solution's derivatives for a mesh of 18 quadratic elements and a bi-cubic analytic solution.

(see [7,8,9]).

Obviously the norm denoted by equation (1) depends upon the choice of the norm on the original space U. Considering for instance the linearized, steady-state Navier-Stokes equations

$$\begin{aligned} -\nu\Delta\boldsymbol{u} + (\boldsymbol{U}\cdot\boldsymbol{\nabla})\boldsymbol{u} + grad\, p &= \boldsymbol{f} \\ div\, \boldsymbol{u} &= 0 \end{aligned} \tag{2}$$

with (for simplicity) Dirichlet boundary conditions on $\boldsymbol{u}$, we define the space U as

$$U = \left\{(\boldsymbol{u},p) \in H_0^1(\Omega) \times L^2(\Omega)\right\} \tag{3}$$

and define the residuals corresponding to the two equations in (2) as

$$\begin{aligned} &\sup_{\boldsymbol{v}\in H_0^1(\Omega),\boldsymbol{v}\neq 0} \int_\Omega \frac{(\nu\boldsymbol{\nabla}u_h\boldsymbol{\nabla}\boldsymbol{v} + (\boldsymbol{U}\cdot\boldsymbol{\nabla})\boldsymbol{u}_h\boldsymbol{v} - p_h div\, \boldsymbol{v} - \boldsymbol{f}\boldsymbol{v})\, d\boldsymbol{x}}{\|\boldsymbol{v}\|} \\ &\sup_{q\in L^2(\Omega),q\neq 0} \frac{\int div\, \boldsymbol{u}_h q d\boldsymbol{x}}{\|q\|} \end{aligned} \tag{4}$$

With the choice of L^2-norm for q, the second of the residuals residuals reduces just to the L^2-residual of the divergence

$$\|div\, \boldsymbol{u}_h\|_{L^2(\Omega)} \tag{5}$$

whereas the choice of the norm for $\boldsymbol{v}$ is neither obvious or unique.

Once a norm is selected for $\boldsymbol{v}$ we may proceed by introducing local Neumann problems over elements Ω_K as

$$\begin{aligned} -\nu\Delta\boldsymbol{\varphi}_K + (\boldsymbol{U}\cdot\boldsymbol{\nabla})\boldsymbol{\varphi}_K &= \boldsymbol{r}_h \\ \boldsymbol{t}(\boldsymbol{\varphi}_k) &= \alpha_K\boldsymbol{g}_h \end{aligned} \tag{6}$$

where $\boldsymbol{r}_h$ is the element residual defined as

$$\boldsymbol{r}_h = \boldsymbol{f} + \nu\Delta\boldsymbol{u}_h - (\boldsymbol{U}\cdot\boldsymbol{\nabla})\boldsymbol{u}_h - grad\, p_h \tag{7}$$

and $\boldsymbol{g}_h$ is the flux residual including the jump in both the pressure (if it is discontinuous) and viscous fluxes (due to the discontinuity of derivatives of $\boldsymbol{u}_h$). The "distribution parameter"

α_K accounts for distributing the flux residual to both neighboring elements in such a way that the local Neumann problems are well-posed.

Once the local problems are solved the element error indicators are defined as

$$\eta_K^2 = \|\varphi_K\|^2 + \|div\, \boldsymbol{u}_h\|^2 \tag{8}$$

This section focuses on the two fundamental issues:

1. How to select the norm.

2. How to determine the distribution parameters α_K.

We emphasize that the use of local Neumann problems does not require any *orthogonality conditions* for the approximate solution $\boldsymbol{u}_h$ and therefore may be applied to $\boldsymbol{u}_h$ resulting from virtually any approximate method.

Error Estimation for Stokes' Flow

One central problem of the project was the development of a general "black box" error estimation package for assessing the accuracy of approximations obtained using finite element, finite difference, or finite volume techniques. Recent theoretical work [10] shows that this goal is now attainable and can be justified on a rigorous mathematical basis, albeit for certain classes of problem.

Unfortunately, as yet, theory is not well developed for the estimation of errors in approximations to the incompressible Navier-Stokes equations:

$$\left.\begin{aligned} -\nu\Delta\boldsymbol{u} + (\boldsymbol{u}\cdot\boldsymbol{\nabla})\boldsymbol{u} + \boldsymbol{\nabla}p &= \boldsymbol{f} \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} &= 0 \end{aligned}\right\} \tag{9}$$

Ultimately, we shall produce an error estimation procedure for these equations. However, rather than attempting to deal with them in their entirety, we propose to focus our attention on the key aspects of this problem by examining several intermediate model problems. The key issues are:

- In what norm should the error be measured? In particular it is necessary to decide on how to "balance" the error in each of the equations or variables $\boldsymbol{u}, p$ appearing in (9).

- How should one deal with the nonlinear term $\boldsymbol{u}\cdot\boldsymbol{\nabla u}$? The question here is not purely related to the nonlinearity but also to the nonsymmetric nature of the term. Once again the question of norms arises. In order to obtain a measure of the error which will be suitable for mesh refinement and enrichment purposes, it is necessary to weight the error upstream since refinements do not only lead to improvement locally but also have an impact downstream.

With these considerations in mind, we shall focus on the following two model problems: *Stokes' Flow*

$$\left.\begin{aligned} -\nu\Delta \boldsymbol{u} + \boldsymbol{\nabla} p &= \boldsymbol{f} \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} &= 0 \end{aligned}\right\} \tag{10}$$

and *Heat Flow With Advection*

$$-\nu\Delta u + (\boldsymbol{a}\cdot\boldsymbol{\nabla})u = f \tag{11}$$

The purpose of studying (10) is to investigate ways of "balancing" errors between the two entities, and by studying (11) one may gain insight into what norm can be used to weight upstream errors correctly.

Stokes' Flow

Let us initially suppose that $\boldsymbol{u}_h$ and p_h are some approximate solution of the Stokes' Problem:

$$\left.\begin{aligned} -\Delta \boldsymbol{u} + \boldsymbol{\nabla} p &= \boldsymbol{f} \\ \boldsymbol{\nabla}\cdot\boldsymbol{u} &= 0 \end{aligned}\right\} \text{ in } \Omega \tag{12}$$

subject to

$$\boldsymbol{u} = \boldsymbol{0} \text{ on } \partial\Omega \tag{13}$$

and

$$\int_\Omega p\, d\boldsymbol{x} = 0 \tag{14}$$

We denote

$$\begin{aligned} X &= H_0^1(\Omega)\times H_0^1(\Omega) \\ M &= L_0^2(\Omega) = \left\{ q\in L^2(\Omega) : \int_\Omega q\, d\boldsymbol{x} = 0 \right\} \end{aligned} \tag{15}$$

The variational form of (12) then becomes:

Find $(\boldsymbol{u}, p) \in X \times M$ such that

$$\left.\begin{aligned} B(\boldsymbol{u},\boldsymbol{v}) \;+\; b(\boldsymbol{v},p) &= (\boldsymbol{f},\boldsymbol{v}) \quad \forall\, \boldsymbol{v} \in X \\ b(\boldsymbol{u},q) &= 0 \qquad\quad \forall\, q \in M \end{aligned}\right\} \tag{16}$$

where

$$\left.\begin{aligned} B(\boldsymbol{u},\boldsymbol{v}) &= \int_\Omega grad\,\boldsymbol{u} : grad\,\boldsymbol{v}\, d\boldsymbol{x} \\ b(\boldsymbol{v},p) &= -\int_\Omega p\, div\,\boldsymbol{v}\, d\boldsymbol{x} \\ (\boldsymbol{f},\boldsymbol{v}) &= \int_\Omega \boldsymbol{f}\cdot\boldsymbol{v}\, d\boldsymbol{x} \end{aligned}\right\} . \tag{17}$$

As a preliminary assumption we assume

$$\boldsymbol{u}_h \in X \;,\; p_h \in M \tag{18}$$

but that otherwise $\boldsymbol{u}_h$ and p_h are arbitrary.

Denote $\boldsymbol{e} = \boldsymbol{u} - \boldsymbol{u}_h \in X$ and $E = p - p_h \in M$. Then from (16) we obtain that $(\mathbf{e}, E) \in X \times M$:

$$\left.\begin{aligned} B(\boldsymbol{e},\boldsymbol{v}) \;+\; b(\boldsymbol{v},E) &= (\boldsymbol{f},\boldsymbol{v}) - B(\boldsymbol{u}_h,\boldsymbol{v}) - b(\boldsymbol{v},p_h)\ \forall\, \boldsymbol{v} \in X \\ b(\boldsymbol{e},q) &= -b(\boldsymbol{u}_h,q) \qquad\qquad \forall\, q \in M \end{aligned}\right\} \tag{19}$$

Balancing of Errors

Let $\varepsilon > 0$ be some user–specified parameter, the interpretation of which will become apparent later in this discussion. Then we define $\left(\hat{\mathbf{e}}, \hat{E}\right) \in X \times M$:

$$\left.\begin{aligned} B\left(\hat{\mathbf{e}},\boldsymbol{v}\right) \;+\; b\left(\boldsymbol{v},\hat{E}\right) &= (\boldsymbol{f},\boldsymbol{v}) - B(\boldsymbol{u}_h,\boldsymbol{v}) - b(\boldsymbol{v},p_h) \quad \forall\, \boldsymbol{v} \in X \\ -\varepsilon\left(\hat{E},q\right) \;+\; b\left(\hat{\mathbf{e}},q\right) &= -b(\boldsymbol{u}_h,q) \quad \forall\, q \in M \end{aligned}\right\} \tag{20}$$

Since $\boldsymbol{v} \in X$ then $\boldsymbol{\nabla}\cdot\boldsymbol{v} \in M$ and so choosing $q = \boldsymbol{\nabla}\cdot\boldsymbol{v}$ in (20) we obtain

$$\begin{aligned} -\ \varepsilon\left(\hat{E},\boldsymbol{\nabla}\cdot\boldsymbol{v}\right) \;+\; b\left(\hat{\mathbf{e}},\boldsymbol{\nabla}\cdot\boldsymbol{v}\right) &= -b\left(\boldsymbol{u}_h,\boldsymbol{\nabla}\cdot\boldsymbol{v}\right) \text{ or} \\ \varepsilon b\left(\boldsymbol{v},\hat{E}\right) \;-\; \left(\boldsymbol{\nabla}\cdot\hat{\mathbf{e}},\boldsymbol{\nabla}\cdot\boldsymbol{v}\right) &= \left(\boldsymbol{\nabla}\cdot\boldsymbol{u}_h,\boldsymbol{\nabla}\cdot\boldsymbol{v}\right) \end{aligned} \tag{21}$$

and now substituting (21) in (20) gives

$$\begin{aligned}\hat{\boldsymbol{e}} \in X : B(\hat{\boldsymbol{e}}, \boldsymbol{v}) + \frac{1}{\varepsilon}(\boldsymbol{\nabla} \cdot \hat{e}, \boldsymbol{\nabla} \cdot \boldsymbol{v}) = (\boldsymbol{f}, \boldsymbol{v}) \quad &- \quad B(\boldsymbol{u}_h, \boldsymbol{v}) - b(\boldsymbol{v}, p_h) \\ &- \quad \frac{1}{\varepsilon}(\boldsymbol{\nabla} \cdot \boldsymbol{u}_h, \boldsymbol{\nabla} \cdot \boldsymbol{v}) \quad \forall\, \boldsymbol{v} \in X\end{aligned} \tag{22}$$

The bilinear form appearing in (22) may be written as

$$\begin{aligned} a(\hat{\boldsymbol{e}}, \boldsymbol{v}) &= B(\hat{\boldsymbol{e}}, \boldsymbol{v}) + \frac{1}{\varepsilon}(\boldsymbol{\nabla} \cdot \hat{\boldsymbol{e}}, \boldsymbol{\nabla} \cdot \boldsymbol{v}) \\ &= \int_\Omega (S\hat{\boldsymbol{e}})^t\, D(S\boldsymbol{v}) d\boldsymbol{x} \end{aligned} \tag{23}$$

where

$$S = \begin{bmatrix} \frac{\partial}{\partial x} & 0 \\ 0 & \frac{\partial}{\partial y} \\ \frac{\partial}{\partial y} & \frac{\partial}{\partial x} \end{bmatrix} \tag{24}$$

and

$$D = \frac{1}{\varepsilon}\begin{bmatrix} 1+\varepsilon & 1-\varepsilon & 0 \\ 1-\varepsilon & 1+\varepsilon & 0 \\ 0 & 0 & \varepsilon \end{bmatrix} \tag{25}$$

With this notation, (22) becomes

$$\hat{\boldsymbol{e}} \in X : a(\hat{\boldsymbol{e}}, \boldsymbol{v}) = (\boldsymbol{f}, \boldsymbol{v}) - a(\boldsymbol{u}_h, \boldsymbol{v}) - b(\boldsymbol{v}, p_h) \quad \forall\, \boldsymbol{v} \in X \tag{26}$$

Formally, the problem (26) is a linear elasticity problem. That is, we can in principle solve (26) to obtain $\hat{\boldsymbol{e}} \approx \boldsymbol{e}$. The natural norm for measuring the solution of (26) is

$$\|\hat{\boldsymbol{e}}\|_E^2 = a(\hat{\boldsymbol{e}}, \hat{\boldsymbol{e}}) = B(\hat{\boldsymbol{e}}, \hat{\boldsymbol{e}}) + \frac{1}{\varepsilon}\|\boldsymbol{\nabla} \cdot \boldsymbol{e}\|_{L_2(\Omega)}^2 \tag{27}$$

and it is with respect to this norm that we propose to assess the accuracy of $\boldsymbol{u}_h$ and indirectly p_h. Notice that the user–specified parameter ε acts as a weight on the divergence of the error.

Thus by selecting an ε, the user specifies how great an effect the violation of incompressibility has on the error. Thus, in order to deal with the Stokes problem it is necessary to first deal with the linear elasticity problem, which is of independent interest also.

Linear Elasticity

Let us consider the following problem:

Find $\boldsymbol{u}$ such that

$$-S^t\boldsymbol{\sigma} = \boldsymbol{f} \text{ in } \Omega, \ \boldsymbol{\sigma} = DS\boldsymbol{u} \tag{28}$$

subject to

$$H\boldsymbol{\sigma} = \boldsymbol{g} \text{ on } \Gamma_N \tag{29}$$

where we suppose the data $\boldsymbol{g}$ and $\boldsymbol{f}$ are compatible and

- D is a general elasticity matrix
- H is the matrix formed using the components of the unit outward pointing normal $\boldsymbol{n} = (n_x, n_y)^T$ as follows

$$H = \begin{bmatrix} n_x & 0 & n_y \\ 0 & n_y & n_x \end{bmatrix} \tag{30}$$

Let

$$X = H^1(\Omega) \times H^1(\Omega)$$

then the weak form of (28)-(29) is:

Find $\boldsymbol{u} \in X$ such that

$$\int_\Omega (S\boldsymbol{u})^t D(S\boldsymbol{v}) d\boldsymbol{x} = \int_\Omega \boldsymbol{f}^t \boldsymbol{v}\, d\boldsymbol{x} + \int_{\Gamma_N} \boldsymbol{g}^t \boldsymbol{v}\, ds \tag{31}$$

for all $\boldsymbol{v} \in X$. Let $\boldsymbol{u}_h \in X$ denote some approximation to the elasticity problem (31) and denote $\boldsymbol{e} = \boldsymbol{u} - \boldsymbol{u}_h$. Then

$$a(\boldsymbol{e}, \boldsymbol{v}) = (\boldsymbol{f}, \boldsymbol{v}) + \langle \boldsymbol{g}, \boldsymbol{v} \rangle_{\Gamma_N} - a(\boldsymbol{u}_h, \boldsymbol{v}) \quad \forall\, \boldsymbol{v} \in X \tag{32}$$

The problem in which we are interested is the estimation, and preferably bounding, of $|||\boldsymbol{e}||| = \sqrt{a(\boldsymbol{e}, \boldsymbol{e})}$. In order to facilitate our analysis it is convenient to introduce a zeroth order term; for $\lambda > 0$ define

$$\hat{a}(\boldsymbol{u},\boldsymbol{v}) = a(\boldsymbol{u},\boldsymbol{v}) + (\lambda \boldsymbol{u},\boldsymbol{v}) \tag{33}$$

The significance of λ will be discussed later. We then let $\hat{\boldsymbol{e}} \in X$:

$$\hat{a}\,(\hat{\boldsymbol{e}},\boldsymbol{v}) = a(\boldsymbol{e},\boldsymbol{v}) \qquad \forall\ \boldsymbol{v} \in X \tag{34}$$

and consider the problem of estimating $\hat{a}\,(\hat{\boldsymbol{e}},\hat{\boldsymbol{e}})$. Obviously as λ is chosen smaller and smaller, we find

$$\lim_{\lambda \to 0^+} \hat{a}\,(\hat{\boldsymbol{e}},\hat{\boldsymbol{e}}) = a(\boldsymbol{e},\boldsymbol{e}) \tag{35}$$

Localization

In order to minimize the cost of obtaining the error estimators, we first of all break (32) into a sequence of local problems on each element.

Let us define various quantities associated with the local problem on Ω_K

$$r_K(\boldsymbol{x}) = \boldsymbol{f}(\boldsymbol{x}) + S^t D S \boldsymbol{u}_h \tag{36}$$

and

$$R_K(S) = \begin{cases} \boldsymbol{g} - HDS\boldsymbol{u}_h & \text{on } \partial\Omega_K \cap \Gamma_N \\ -\alpha_{KL}[\![HDS\boldsymbol{u}_h]\!] & \text{on } \partial\Omega_K \cap \partial\Omega_L \end{cases} \tag{37}$$

where $[\![\cdot]\!]$ denotes the jump in tractions of the approximate solution across the interelement boundary. Here α_{KL} is a 3×3 diagonal matrix weighting of the approximate tractions from elements Ω_K and Ω_L on the edge of $\Gamma_{KL} = \partial\Omega_K \cap \partial\Omega_L$. The weighting should ideally be chosen so that

$$\langle HDS\boldsymbol{u}_h \rangle_\alpha := \alpha_{KL}\,(HDS\boldsymbol{u}_h)_K + \ \alpha_{LK}\,(HDS\boldsymbol{u}_h)_L \approx HDS\boldsymbol{u} \tag{38}$$

(i.e., the weighted average of the tractions) represents a good approximation to the true traction. Obviously, in order to obtain consistency we require

$$\alpha_{KL} + \alpha_{LK} = I \tag{39}$$

where I is the 3×3 identity matrix. In most applications to date, the choice $\alpha_{KL} \equiv \frac{1}{2} I$ was sacrosanct, but it can be shown [10] that this choice has severe repercussions regarding the accuracy of the resulting estimator. The local problems then become

Find $\boldsymbol{\varphi}_K \in H^1(\Omega_K) \times H^1(\Omega_K)$: for all $\boldsymbol{\omega} \in H^1(\Omega_K) \times H^1(\Omega_K)$,

$$\begin{aligned} \int_{\Omega_K} (S\boldsymbol{\varphi}_K)^t D(S\boldsymbol{\omega}) d\boldsymbol{x} &= \int_{\Omega_K} \boldsymbol{r}_K(\boldsymbol{x})^t \boldsymbol{\omega} d\boldsymbol{x} + \oint_{\partial\Omega_K} \boldsymbol{R}_K(s)^t \boldsymbol{\omega}(s) ds \\ &- \boldsymbol{\delta}_K^t \int_{\Omega_K} \boldsymbol{\omega} d\boldsymbol{x} \end{aligned} \tag{40}$$

where

$$\boldsymbol{\delta}_K = \frac{1}{|\Omega_K|} \left\{ \int_{\Omega_K} \boldsymbol{r}_K(\boldsymbol{x}) d\boldsymbol{x} + \oint_{\partial\Omega_K} \boldsymbol{R}_K(s) ds \right\} \tag{41}$$

The physical interpretation of the term involving $\boldsymbol{\delta}_K$, which arises naturally from the mathematical analysis, is to enforce equilibrium on the local problem. Having obtained $\boldsymbol{\varphi}_K$ we then define

$$\eta_K^2 = \int_{\Omega_K} (S\boldsymbol{\varphi}_K)^t D(S\boldsymbol{\varphi}_K) d\boldsymbol{x} + \frac{|\Omega_K|}{\lambda} |\boldsymbol{\delta}_K|^2 \tag{42}$$

The following result, generalizing a similar result in [10], can be shown:

$$|||\hat{e}|||^2 + \lambda \, ||\hat{e}||^2 \leq \sum_{\Omega_K} \eta_k^2 \tag{43}$$

Of course this gives an upper bound on $\hat{a}(\hat{e}, \hat{e})$. However, let us suppose that we can choose α_{KL} (see (37)–(39)) in such a way that $\delta_K \equiv 0$. Then letting $\lambda \to 0^+$ and using (33), (42), and (43) gives

$$|||e|||^2 \leq \sum_{\Omega_K} \int_{\Omega_K} (S\boldsymbol{\varphi}_K)^t D(S\boldsymbol{\varphi}_K) dx \tag{44}$$

This result is perhaps rather astonishing, proclaiming an *upper bound* on the discretization error. The key is the assumption that $\boldsymbol{\delta}_K$ can be driven to zero. This may be understood intuitively as follows. In order to estimate the error we can essentially only base our observations on the residuals $\boldsymbol{r}_K$ and $\boldsymbol{R}_K$. However, there is a third type of residual, hitherto neglected namely that the true solution is in equilibrium *locally.* This corresponds to $\boldsymbol{\delta}_K \equiv \mathbf{0}$, therefore we should incorporate $\boldsymbol{\delta}_K$ into our analysis of the error. The above discussion shows how to accomplish this.

Flux Splitting

We now make some remarks on how α_{KL} should be chosen to ensure $\boldsymbol{\delta}_K = \mathbf{0}$. The most crude, but simplest, way to achieve this is by solving the quadratic programming problem

$$\min \sum_{\Omega_K} |\boldsymbol{\delta}_K|^2 \text{ subject to } \alpha_{KL} + \alpha_{LK} = I \text{ on } \partial\Omega_K \cap \partial\Omega_L \tag{45}$$

This represents a computation on a global scale and is therefore rather expensive. However, a sensible strategy would be to choose $\alpha_{KL} \equiv \frac{1}{2}I$ during the adaptive analysis and then perform (45) only once in the final phase when the final solution has been obtained. The main drawback of this technique is the expense. However, if we assume that the approximation arises from an *h-p* element computation then considerable savings result.

Flux Splitting for *h-p* Finite Element Computations

Let us now assume $\boldsymbol{u}_h$ is an *h-p* finite element approximation. In [10] it was shown that α_{KL} for which $\boldsymbol{\delta}_K$ vanishes can be calculated by *local* computations only. This makes the cost very inexpensive and is suitable for parallelization, reducing the time to a negligible amount. We shall not give the precise details at this point but it is worth noting that the only computations required are the solutions of small (at most 6 × 6 in two dimensions) systems of algebraic equations.

3.2.3 Error Estimation Using the Flux Balancing Technique

In this section we provide a more complete discussion for the flux balancing technique for equilibrating the local problems during error estimation. For additional details on this technique, we refer to Ainsworth and Oden [11]. Although the following discussion is presented in the context of finite element techniques, the approximate solution whose error is being estimated could have been calculated using almost any approximation technique. In this case, the finite elements merely represent computational cells. This flux balancing technique was implemented into a two-dimensional finite element test code and used to compute error estimates on several classes of problems. After the following discussion, we present the results of sample computations performed for a vector-valued model problem.

Self-Equilibrating Flux Splittings

First we begin with some historical remarks from which the idea of self-equilibrating flux splittings evolved. Consider for a moment the Neumann problem:

Find w such that

$$-\Delta w(\boldsymbol{x}) = f(\boldsymbol{x}) \text{ in } \Omega \text{ subject to } \frac{\partial w}{\partial n} = g \text{ on } \partial\Omega \tag{46}$$

The well-known condition for the existence of solutions to this problem is that

$$\int_\Omega f(\boldsymbol{x})d\boldsymbol{x} + \oint_{\partial\Omega} g(s)ds = 0 \tag{47}$$

When considering the problem of obtaining *a posteriori* estimates of the error in a finite element approximation of (46), one is led to consider local Neumann problems on each element of the partition with data related to the residual $f + \Delta\hat{u}$ and to the jumps in the normal fluxes across interelement boundaries. Of course it is necessary that the data satisfy the compatibility condition (47) in order for the local problem to be well posed.

In general the compatibility condition will be violated. Various *ad hoc* ways of reformulating the local error problem have been used to circumvent this incompatibility problem. However, Kelly [12] proposed that the fluxes be split unequally between elements in such a way that the compatibility condition be satisfied, referring to this idea as self–equilibration.

In one dimension for a linear approximation, the splitting could be found explicitly in terms of the interelement jumps in the normal fluxes and the residual integrated against the basis functions. However, in higher dimensions such an explicit formula could not be found and instead Kelly resorted to minimizing the sum over all the elements of the squares of the amounts by which the compatibility condition is violated. It was found that splittings could be computed for which the lack of equilibrium was negligible.

Kelly [12] gives numerical results demonstrating the significant improvement in performance of the error estimators when the self–equilibration is applied. However, the idea has not been widely adopted by the finite element community, in spite of the increased emphasis on *a posteriori* error estimation. The primary reasons are probably because the procedure is extremely costly (due to a global minimization) and is needed only for operators such as the Laplacian.

Generalized Flux Splitting

More recently, once again working from the viewpoint of *a posteriori* error estimation, it has been shown [10] that a type of equilibration condition has a significant impact on the efficiency of *a posteriori* error estimators for more general second order differential operators. In order to illustrate the idea let us consider the simple example:

Find w such that

$$-\Delta w(\boldsymbol{x}) + w(\boldsymbol{x}) = f(\boldsymbol{x}) \text{ in } \Omega \text{ subject to } \frac{\partial w}{\partial n} = g \text{ on } \partial\Omega \tag{48}$$

The issue here is not the existence of solutions to the problem since the problem above is always well posed. However, when one is attempting to compute *a posteriori* error estimators for the problem, once again a local problem on each element K of the same form arises with data related to the residual $f + \Delta\hat{w} - \hat{w}$ and the jumps in the normal fluxes across the interelement boundaries. In [10] it is shown that the resulting estimator can be rather pessimistic if the data for the local problem is not chosen with some care. Quantitatively, this statement means that the data g_K for the local problem should be chosen such that

$$\int_K \big(f(\boldsymbol{x}) - \hat{w}(\boldsymbol{x})\big)d\boldsymbol{x} + \oint_{\partial K} g_K(s)ds = 0 \tag{49}$$

or at any rate such that this quantity is sufficiently small. If this were the only constraint on g_K then there would be little difficulty. However, there is an additional constraint between g_K and the flux g_L on the shared side Γ_{KL} of the neighboring element which destroys the local nature of the condition.

Once again, the problem is to split the fluxes on the interelement side between the two neighboring elements in such a way as to satisfy a type of generalized equilibration condition.

Flux Recovery Scheme

In this section we give a heuristic alternative interpretation of the flux splitting discussed above. Flux recovery and postprocessing schemes have been intensively studied [13] for the finite element method on uniform meshes for problems with smooth solutions.

The strategy used in this work to obtain enhanced approximations to the flux was to average the fluxes along the interelement edges. Since the meshes were assumed to be essentially uniform and of uniform degree by symmetry, one should take the average of the two possible flux approximations along the edge. Indeed, one often finds that this type of scheme yields superconvergent approximations.

For nonuniform meshes, however, a weighted average should be chosen instead of a simple average. Also, in order to effectively deal with non-smooth functions, the weights must be allowed to depend on the approximation *itself*, giving a *nonlinear* postprocessing.

The question then remains as to how these weights should be chosen. Since the true solution satisfies (48), integrating over an element K gives

$$\int_K \Big(f(\boldsymbol{x}) - w(\boldsymbol{x})\Big)d\boldsymbol{x} + \oint_{\partial K} \frac{\partial w}{\partial n_K} ds = 0 \tag{50}$$

Therefore, one might try to choose the approximation to the flux g_K in such a way that

$$\int_K \Big(f(\boldsymbol{x}) - \hat{w}(\boldsymbol{x})\Big)d\boldsymbol{x} + \oint_{\partial K} g_K(s)ds = 0 \tag{51}$$

which is the same condition as was arrived at in the previous section.

Notation and Preliminaries

In order to simplify the presentation, we shall consider the two-dimensional case. The extension to three dimensions will become clear after a few remarks of clarification.

Let $\Omega \subset \mathbb{R}^2$ be an open bounded domain with boundary Γ consisting of a finite number of smooth arcs meeting with internal angle $\theta \in (0, 2\pi)$. $H^m(\Omega)$, $m \in Z^+$ will be used to denote the standard Sobolev space equipped with the norm $\|u\|_{m,\Omega}$. We use the notation $H^0(\Omega) = L_2(\Omega)$ in the case $m = 0$.

Let $\mathcal{P}$ denote a partitioning of Ω into a collection of $N = N(\mathcal{P})$ subdomains K such that

1. $N(\mathcal{P}) < \infty$

2. $\overline{\Omega} = \bigcup_{\rightarrow k} \overline{K}$

3. If $K \neq L$ than $K \cap L$ is empty

4. K are Lipschitzian domains with piecewise smooth boundaries ∂K

5. $\Gamma_{KL} = \partial K \cap \partial L$ is an entire edge of at least one of K or L

It is convenient to formally define the exterior of Ω to be the zeroth element. In this way, the complete set of element edges may be characterized by

$$E = E(\mathcal{P}) = \bigcup_{\rightarrow K,L:K>L\geq 0} \Gamma_{KL} \tag{52}$$

The set of interelement edges may be characterized by

$$E_I = E_I(\mathcal{P}) = \bigcup_{\rightarrow K,L:K>L>0} \Gamma_{KL} \tag{53}$$

The unit outward pointing normal vector on K is denoted by $\boldsymbol{n}_K$. Let

$$\sigma_{KL} = -\sigma_{LK} = \begin{cases} +1, & \text{if} \quad K > L, \\ -1, & \text{if} \quad K < L \end{cases} \tag{54}$$

and define

$$\boldsymbol{n}(s) = \sigma_{KL}\boldsymbol{n}_K(s) = \sigma_{LK}\boldsymbol{n}_L(s), \qquad s \in \Gamma_{KL} \tag{55}$$

That is, $\boldsymbol{n}$ points outward from the domain with the largest index.

We now focus our attention on a generalized model problem. Let n be the number of components of the problem and $\chi = [H^1(\Omega)]^n$. Let $\mathcal{B}: \chi \times \chi \rightarrow \mathbb{R}$ denote the bilinear form

$$\mathcal{B}(\boldsymbol{u}, \boldsymbol{v}) = \sum_{j,k=1}^{n} B^{jk}(u_j, v_k) \tag{56}$$

where $\boldsymbol{u} = (u_1, \ldots, u_n)$, $\boldsymbol{v} = (v_1, \ldots, v_n)$ and

$$\begin{aligned} B^{jk}(u_j, v_k) &= \int_\Omega \left\{\nabla v_k \cdot A^{kj}\nabla u_j + v_k B^{kj} \cdot \nabla u_j + v_k C^{kj} u_j + \nabla v_k \cdot D^{kj} u_j\right\} d\boldsymbol{x} \\ &+ \int_{\partial\Omega} \left\{v_k b_s^{kj} \frac{\partial u_j}{\partial s} + v_k c_s^{kj} u_j\right\} ds \end{aligned} \tag{57}$$

where $A^{kj} \in \mathbb{R}^{2\times 2}$, $B^{kj} \in \mathbb{R}^{2\times 1}$, $D^{kj} \in \mathbb{R}^{1\times 2}$, and C^{kj}, b_s^{kj}, and $c_s^{kj} \in \mathbb{R}$. We shall assume that the bilinear form is continuous and coercive on $\chi \times \chi$, that is to say, there exist positive constants M and α such that

$$\mathcal{B}(\boldsymbol{u},\boldsymbol{v}) \leq M \left\|\boldsymbol{u}\right\|_\chi \left\|\boldsymbol{v}\right\|_\chi \qquad \forall\ \boldsymbol{u},\boldsymbol{v} \in \chi \tag{58}$$

and

$$\mathcal{B}(\boldsymbol{v},\boldsymbol{v}) \geq \alpha \left\|\boldsymbol{v}\right\|_\chi^2 \qquad \forall\ \boldsymbol{v} \in \chi \tag{59}$$

Let $\mathcal{L}: \chi \rightarrow \mathbb{R}$ denote the linear form

$$\mathcal{L}(\boldsymbol{v}) = \sum_{k=1}^{n} L^k\left(v_k\right) \tag{60}$$

where

$$L^k\left(v_k\right) = \int_\Omega f^k v_k d\boldsymbol{x} + \int_{\partial\Omega} f_s^k v_k ds \tag{61}$$

where f^k and $f_s^k \in \mathbb{R}$. We shall assume that the linear form is continuous on χ.

Under the above assumptions there exists a unique solution $\boldsymbol{u} \in \chi$ of the problem:

Find $\boldsymbol{u} \in \chi$ such that

$$\mathcal{B}(\boldsymbol{u},\boldsymbol{v}) = \mathcal{L}(\boldsymbol{v}) \qquad \forall\ \boldsymbol{v} \in \chi \tag{62}$$

Let $\hat{\chi} \subset \chi$ denote a finite dimensional subspace consisting of continuous piecewise polynomial functions defined on the partition $\mathcal{P}$. The polynomial degree is allowed to vary from element to element but the functions are constrained in such a way that interelement continuity is preserved.

Our error estimation problem consists of estimating the error in an approximate solution $\hat{\boldsymbol{u}} \in \hat{\chi}$ where $\hat{\boldsymbol{u}}$ is obtained using some approximation technique. It will be useful to consider the structure of the space $\hat{\chi}$ more carefully. In particular, let A be any *unconstrained* or *proper* vertex A in the partition. Associated with each such node is a piecewise polynomial ψ_A of degree one, which vanishes at every other regular node in the partition and (subject to suitable scaling) has the value 1 at the node A. For k-irregular meshes [14], the support of ψ_A consists of a patch of elements containing or near to the node A. In the case of regular meshes, the support is merely the set of elements of which A is a vertex.

Let $\mathcal{F}(\mathcal{P})$ denote the set of all unconstrained nodes in the partition. It is readily seen that the set $\{\psi_A\}_{A\in\mathcal{F}}$ forms a partition of unity subordinate to the covering $\{K\}$ on the domain Ω. That is to say

$$\sum_{A \in \mathcal{F}(\mathcal{P})} \psi_A(\boldsymbol{x}) \equiv 1, \qquad \forall\, \boldsymbol{x} \in \Omega \tag{63}$$

Let us note that (62) implies *a fortiori*

$$\sum_{j=1}^{n} B^{jk}(u_j, \psi_A) = L^k(\psi_A) \tag{64}$$

for $k = 1, \ldots, n$ and any $A \in \mathcal{F}(\mathcal{P})$.

For each element $K \in \mathcal{P}$ and for $j = 1, \ldots, n$ let

$$Q^k_K(\boldsymbol{x}) = \left\{ \sum_{j=1}^{n} \left(A^{kj} \nabla u_j + D^{kj} u_j \right) \right\} \Bigg|_K \tag{65}$$

i.e., Q^k_K denotes the k-th component of the flux on element K. With each interelement edge $\Gamma_{KL} \in E(\mathcal{P})$ we associate n functions $\alpha^{(k)}_{KL}(s)$, $s \in \Gamma_{KL}$ corresponding to each component of the flux. We shall distinguish between the $\alpha^{(k)}_{KL}$ and $\alpha^{(k)}_{LK}$, and usually these will be unequal, but will be required to satisfy the condition

$$\alpha^{(k)}_{KL}(s) + \alpha^{(k)}_{LK}(s) = 1, \qquad s \in \Gamma_{KL} \tag{66}$$

The normal component of the flux will usually be discontinuous across the interelement edges. The jump in the flux is denoted by

$$\left[\!\left[\boldsymbol{n} \cdot Q^k \right]\!\right](s) = \boldsymbol{n}_K \cdot \left(Q^k_K - Q^k_L \right) = \boldsymbol{n}_K \cdot Q^k_K + \boldsymbol{n}_L \cdot Q^k_L, \qquad s \in \Gamma_{KL} \tag{67}$$

The $\alpha^{(k)}_{KL}$ are used to construct an average normal flux along the Γ_{KL} from Q^k_K and Q^k_L as follows

$$\left\langle \boldsymbol{n} \cdot Q^k \right\rangle_{1-\alpha} = \left(\alpha^{(k)}_{LK} Q^k_K + \alpha^{(k)}_{KL} Q^k_L \right) \cdot \boldsymbol{n} \tag{68}$$

Remark: *The subscript is written as $1 - \alpha$ to emphasize the fact that the average is formed using α_{LK} times the flux from element K rather than the more natural notational choice of α_{KL}.*

Existence of Self–Equilibrating Flux Splittings

The preceding remarks indicate some of the reasons for obtaining self–equilibrating flux splittings. However, the question of whether such splittings exist arises. In this section we prove that such splittings can be found.

First of all, we formulate the appropriate equilibration condition for the model system (62). The strong form of (62) is to find $(u_1, \ldots, u_n)$ such that for $k = 1, \ldots, n$,

$$\sum_{j=1}^{n}\left\{-\nabla\cdot\left(A^{kj}\nabla u_j\right)+B^{kj}\cdot\nabla u_j+C^{kj}u_j-\nabla\cdot\left(D^{kj}u_j\right)\right\}=f^k \text{ in } \Omega \tag{69}$$

and subject to the boundary conditions for $k=1,\ldots,n$

$$\sum_{j=1}^{n}\left\{\boldsymbol{n}\cdot\left(A^{kj}\nabla u_j+D^{kj}u_j\right)+b_s^{kj}\frac{\partial u_j}{\partial s}+c_s^{kj}u_j\right\}=f_s^k \text{ on } \partial\Omega \tag{70}$$

Let $K\in\mathcal{P}$ and let ϕ be sufficiently smooth on K, then multiplying (69) by ϕ, integrating over K and applying the boundary condition (70) on $\partial K\cap\partial\Omega$, gives for $k=1,\ldots,n$

$$\sum_{j=1}^{n}B_K^{jk}\left(u_j,\phi\right)=L_K^k(\phi)+\int_{\partial K\backslash\partial\Omega}\phi\boldsymbol{n}_K\cdot\sum_{j=1}^{n}\left(A^{kj}\nabla u_j+D^{kj}u_j\right)ds \tag{71}$$

and now choosing $\phi\equiv 1$ gives

$$\sum_{j=1}^{n}B_K^{jk}\left(u_j,1\right)=L_K^k(1)+\int_{\partial K\backslash\partial\Omega}\boldsymbol{n}_K\cdot\sum_{j=1}^{n}\left(A^{kj}\nabla u_j+D^{kj}u_j\right)ds \tag{72}$$

This equation expresses the equilibration of the true solution to the model problem. The equilibration condition for approximation $\hat{\boldsymbol{u}}$ and hence for the error estimation problem is correspondingly

$$\sum_{j=1}^{n}B_K^{jk}\left(\hat{u}_j,1\right)=L_K^k(1)+\int_{\partial K\backslash\partial\Omega}\left\langle\boldsymbol{n}_K\cdot Q^k\right\rangle_{1-\alpha}ds \tag{73}$$

Although the above derivation is rather informal, the result can be obtained rigorously by following the corresponding steps for the scalar case in [10].

Let $\Delta_K^{(k)}$ denote the lack of equilibration in the k-th component on element K, that is

$$\Delta_K^{(k)}=L_K^k(1)-\sum_{j=1}^{n}B_K^{jk}\left(\hat{u}_j,1\right)+\int_{\partial K\backslash\partial\Omega}\left\langle\boldsymbol{n}_K\cdot Q^k\right\rangle_{1-\alpha}ds \tag{74}$$

The remainder of this section is concerned with proving that there exist choices of α for which the equilibration condition can be satisfied on every element $K\in\mathcal{P}$ and all $k=1,\ldots,n$.

The replacement of unity in (74) with the partition given by (63) yields

$$\Delta_K^{(k)}=\sum_A\Delta_{K,A}^{(k)} \tag{75}$$

where

$$\Delta_{K,A}^{(k)}=L_K^k\left(\psi_A\right)-\sum_{j=1}^{n}B_K^{jk}\left(\hat{u}_j,\psi_A\right)+\int_{\partial K\backslash\partial\Omega}\psi_A(s)\left\langle\boldsymbol{n}_K\cdot Q^k\right\rangle_{1-\alpha}ds \tag{76}$$

Therefore, a sufficient condition for equilibration is to ensure that $\Delta_{K,A}^{(k)}$ vanishes for all K, A, and k. Rather than work with $\alpha_{KL}^{(k)}$ directly, it will be convenient to introduce the following quantities

$$\mu_{KL}^{(k)} = \alpha_{KL}^{(k)} - \frac{1}{2} \text{ and } \mu_{LK}^{(k)} = \alpha_{LK}^{(k)} - \frac{1}{2} \tag{77}$$

so that using (66), we obtain that μ is anti-symmetric:

$$\mu_{KL}^{(k)} = -\mu_{LK}^{(k)} \tag{78}$$

Using the new variables, one finds that

$$\left\langle \boldsymbol{n}_K \cdot Q^k \right\rangle_{1-\alpha} = \left\langle \boldsymbol{n}_K \cdot Q^k \right\rangle_{\frac{1}{2}} + \mu_{LK}^{(k)} \left[\!\left[\boldsymbol{n} \cdot Q^K \right]\!\right] \tag{79}$$

and that

$$\begin{aligned} \Delta_{K,A}^{(k)} &= L_K^k(\psi_A) - \sum_{j=1}^{n} B_K^{jk}(\hat{u}_j, \psi_A) \\ &+ \int_{\partial K \setminus \partial\Omega} \psi_A(s) \left\langle \boldsymbol{n}_K \cdot Q^k \right\rangle_{\frac{1}{2}} ds + \sum_{L>0} \int_{\Gamma_{KL}} \psi_A(s) \mu_{KL}^{(k)} \left[\!\left[\boldsymbol{n}_K \cdot Q^k \right]\!\right] ds \end{aligned} \tag{80}$$

Denote

$$b_{K,A}^{(k)} = L_K^k(\psi_A) - \sum_{j=1}^{n} B_K^{jk}(\hat{u}_j, \psi_A) + \int_{\partial K \setminus \partial\Omega} \psi_A(s) \left\langle \boldsymbol{n}_K \cdot Q^k \right\rangle_{\frac{1}{2}} ds \tag{81}$$

and

$$\rho_{LK,A}^{(k)} = -\int_{\Gamma_{KL}} \psi_A(s) \left[\!\left[\boldsymbol{n}_K \cdot Q^k \right]\!\right] ds \tag{82}$$

then the vanishing of $\Delta_{K,A}^{(k)}$ is equivalent to

$$\sum_{L>0} \mu_{LK}^{(k)} \rho_{LK,A}^{(k)} = b_{K,A}^{(k)} \tag{83}$$

Let $\hat{\mu}_{LK}^{(k)} = \mu_{LK}^{(k)} \rho_{LK,A}^{(k)}$ then we obtain

$$\sum_{L>0} \hat{\mu}_{LK}^{(k)} = b_{K,A}^{(k)} \tag{84}$$

and using (78) to eliminate $\hat{\mu}_{KL}^{(k)}$ for $L > K$ gives

$$\sum_{K>L>0} \left(\hat{\mu}^{(k)}_{LK} - \hat{\mu}^{(k)}_{KL}\right) = b^{(k)}_{K,A} \tag{85}$$

which may be written in the form of a sequence of linear systems of equations

$$M_A \hat{\mu}^{(k)} = b^{(k)}_A \tag{86}$$

for $k = 1, \ldots, n$. The underlying Matrix M_A depends only on the topology of the patch of elements forming the support of ψ_A, and not on the index k. We shall return to this point later since it has a significant impact on the cost of the algorithm for computing the splittings.

The structure of the matrix M_A was studied in [10] where it was shown that the kernel of the adjoint matrix M_A^* was the one-dimensional subspace spanned by vectors of the form $(1, \ldots, 1)$. Thus a necessary and sufficient condition for the existence of solutions to the system (86) is that for $k = 1, \ldots, n$:

$$\sum_{K \in \mathcal{P}} b^{(k)}_{K,A} = 0 \tag{87}$$

Now for the special case where $\hat{\boldsymbol{u}}$ satisfies

$$\mathcal{B}(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}) = \mathcal{L}(\hat{\boldsymbol{v}}) \qquad \forall\, \hat{\boldsymbol{v}} \in \hat{\mathcal{X}} \tag{88}$$

then

$$\begin{aligned}
\sum_{K \in \mathcal{P}} b^{(k)}_{K,A} &= \sum_{K \in \mathcal{P}} \left\{ L^k_K(\psi_A) - \sum_{j=1}^{n} B^{jk}_K(\hat{u}_j, \psi_A) + \int_{\partial K \backslash \partial\Omega} \psi_A(s) \left\langle \boldsymbol{n}_K \cdot Q^k \right\rangle_{\frac{1}{2}} ds \right\} \\
&= L^k(\psi_A) - \sum_{j=1}^{n} B^{jk}(\hat{u}_j, \psi_A) + \sum_{K \in \mathcal{P}} \int_{\partial K \backslash \partial\Omega} \psi_A(s) \left\langle \boldsymbol{n}_K \cdot Q^k \right\rangle_{\frac{1}{2}} ds \\
&= \sum_{K \in \mathcal{P}} \int_{\partial K \backslash \partial\Omega} \psi_A(s) \left\langle \boldsymbol{n}_K \cdot Q^k \right\rangle_{\frac{1}{2}} ds \\
&= \sum_{K \in \mathcal{P}} \sum_{L \in \mathcal{P}} \int_{\Gamma_{KL}} \psi_A(s) \sigma_{KL} \left\langle \boldsymbol{n} \cdot Q^k \right\rangle_{\frac{1}{2}} ds \\
&= 0
\end{aligned} \tag{89}$$

where we have made use of (68) and (54). Consequently, for this case, the systems (86) always have a solution which is unique up to the addition of an arbitrary multiple of the vector $(1,\ldots,1)$. Consequently, we have shown that $\Delta^{(k)}_{K,A}$ can be made to vanish and thus that there exist choices of α for which the equilibration condition is satisfied.

An Algorithm for Flux Splitting

The principal results of the previous section are that the self-equilibrating flux splittings can be constructed directly using solutions of the systems (86). The matrix M_A does not depend on a particular component of the system, but only on the topology of the patch formed by the elements constituting the support of the linear basis function ψ_A. Therefore, for the purposes of this section we can dispense with the superscript k referring to the component of the system.

Without loss of generality, we suppose the elements forming the patch to be numbered from 1 to N_A. The matrix M_A also depends on the interelement edges between the elements forming the patch. Let us arbitrarily label these edges from 1 to N_E. Then, M_A has one row corresponding to each of the elements $1,\ldots,N_A$ and one column for each of the edges $\Gamma_1,\ldots,\Gamma_{N_E}$. Examining the equations leading to the linear system (86), we see that the structure of M_A is

$$(M_A)_{ij} = \begin{cases} 0 & \text{if} \quad \partial\Omega_i \cap \partial\Omega_k \text{ is empty for all } k, \\ 1 & \text{if} \quad \partial\Omega_i \cap \partial\Omega_k = \Gamma_j \text{ and } k < i \\ -1 & \text{if} \quad \partial\Omega_i \cap \partial\Omega_k = \Gamma_j \text{ and } k > i \end{cases} \tag{90}$$

The matrix M_A is singular, meaning that the solution $\widehat{\mu}$ of

$$M_A \widehat{\mu} = b_A \tag{91}$$

is unique only up to the addition of a multiple of $(1,\ldots,1)$. From a theoretical point of view it is irrelevant which solution of this system we pick. However, from a practical viewpoint, one should choose the smallest solution since this will inhibit the effect of rounding errors. Therefore, let us try to find $\widehat{\mu}$ such that $\widehat{\mu}^T\widehat{\mu}$ is minimized subject to (91). Let $\mathcal{Y}$ denote the Lagrangian

$$\mathcal{Y}(\widehat{\mu},\lambda) = \frac{1}{2}\widehat{\mu}^T\widehat{\mu} - (M_A\widehat{\mu} - b_A)^T\lambda \tag{92}$$

then the conditions for a stationary point are

$$\widehat{\mu} = M_A^T \lambda \tag{93}$$

and

$$M_A \widehat{\mu} = b_A \tag{94}$$

Together these conditions imply that

$$M_A M_A^T \lambda = b_A \tag{95}$$

Therefore we may obtain $\widehat{\mu}$ by solving (95) and then using (94) to compute $\widehat{\mu}$. At first sight this process seems to offer little over attempting to solve (91) directly. However, the special structure of the matrix T_A defined by

$$T_A = M_A M_A^T \tag{96}$$

means that there are advantages in pursuing this approach.

Structure of T_A

First, notice that T_A is symmetric and positive semi-definite. The kernel of T_A is identical to the kernel of M_A. Making use of (90) we readily conclude that the elements of T_A are given by

$$(T_A)_{ij} = \begin{cases} C_i & \text{if } j = 1 \\ -1 & \text{if } \Omega_i \text{ and } \Omega_k \text{ share a common edge,} \\ 0 & \text{otherwise} \end{cases} \tag{97}$$

where C_i is the number of elements in the patch which share an edge with element Ω_i. It is apparent that T_A can be rapidly constructed purely from the topology of the patch of elements forming the support of ψ_A, and for this reason we often refer to T_A as the *topology matrix* for the node A. Some examples of topology matrices for various types of patches are shown in Figs. 3.2 and 3.3.

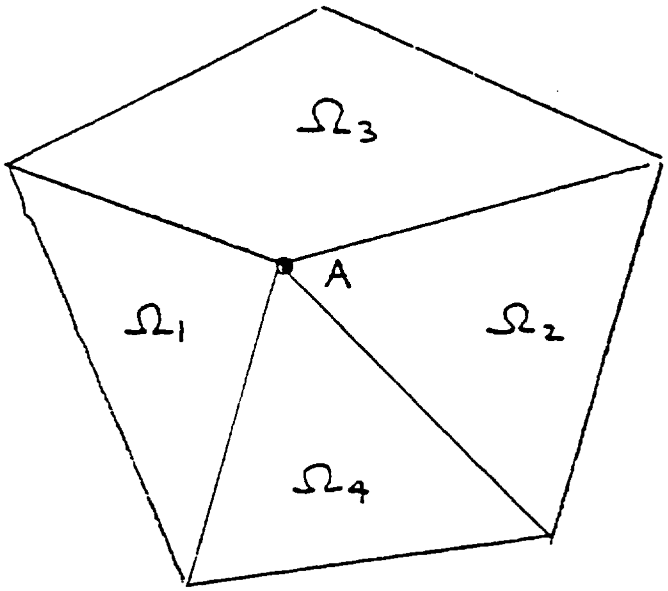


Figure 3.2: Topology of matrix for interior node on regular mesh.

$$T_A = \begin{bmatrix} 2 & 0 & -1 & -1 \\ 0 & 2 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{bmatrix}$$

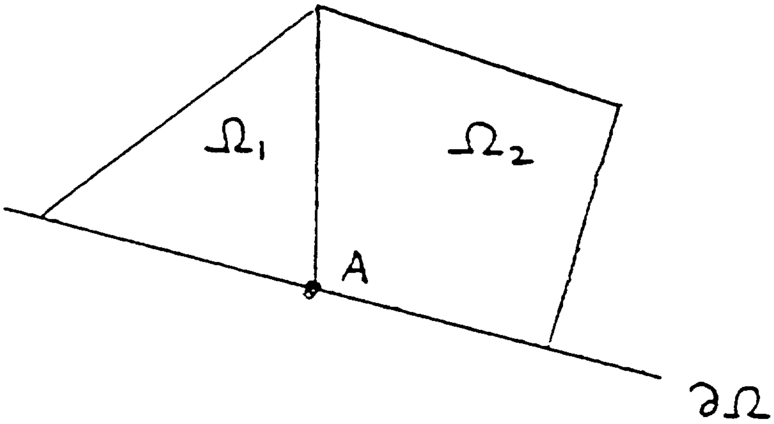


Figure 3.3: Topology matrix for boundary node.

$$T_A = \begin{bmatrix} 1 & -1 \\ -1 & 1 \end{bmatrix}$$

Solution of $T_A \hat{\mu} = b_A$

The matrix T_A is still singular and the same considerations mentioned earlier still apply. Moreover, since $\hat{\boldsymbol{u}}$ may not satisfy (88) or due to rounding error and the familiar *variational crimes* [14], the right hand sides $b_A^{(k)}$ may not in fact satisfy the condition (87) precisely, meaning that the system actually seen by the computer is singular and has no solutions.

In order to combat this type of difficulty we apply a two-step procedure: First, in order to ensure that the matrix equation has a solution, we modify $b_A^{(k)}$ according to

$$b_{K,A}^{(k)} \leftarrow b_{K,A}^{(k)} - \delta_A^{(k)} \tag{98}$$

for $K = 1, \ldots, N_A$, where

$$\delta_A^{(k)} = \frac{1}{N_A} \sum_{K=1}^{N_A} b_{K,A}^{(k)} \tag{99}$$

This process means that the condition (87) is enforced while avoiding destroying too much of the information present in the components of $b_A^{(k)}$. Second, in order to make the matrix equation easier to solve and to obtain a reasonably small solution from the set of solution vectors, we modify T_A to become

$$T_A \leftarrow T_A + D \tag{100}$$

where

$$D = \begin{bmatrix} 1 & 1 & \cdots & 1 \\ 1 & 1 & \cdots & 1 \\ \vdots & \vdots & \ddots & \vdots \\ 1 & 1 & \cdots & 1 \end{bmatrix} \tag{101}$$

This does not change the essential properties of the matrix owing to the condition (87) satisfied by the data, but merely picks a particular solution. However, it does mean that the matrix equation is easier to solve since it is now nonsingular. An alternative method for picking out a particular solution would be to set, for example $(T_A)_{11}$ to some large value. This is similar to the "big spring" method used in finite element analysis. However, this would completely destroy the information present in the first component of $b_A^{(k)}$. The method chosen balances the information which is lost between all the components of $b_A^{(k)}$.

To see that the matrix is nonsingular, let $\boldsymbol{x} \in \mathbb{R}^{N_A}$ denote any vector. Let $V = \text{span}(1, \ldots, 1)$ and $V^{\perp}$ be the orthogonal complement with respect to the Euclidian inner product on $\mathbb{R}^{N_A}$. This $\boldsymbol{x}$ may be written as $\boldsymbol{y} + \boldsymbol{z}$ where $\boldsymbol{y} \in V$ is the orthogonal projection of $\boldsymbol{x}$ onto V and $\boldsymbol{z} = \boldsymbol{x} - \boldsymbol{y} \in V^{\perp}$. Using these definitions it follows that

$$\begin{aligned} & \boldsymbol{x}^T (T_A + D) \boldsymbol{x} \\ = & \ \boldsymbol{x}^T (T_A + D)(\boldsymbol{y} + \boldsymbol{z}) \\ = & \ \boldsymbol{x}^T (T_A \boldsymbol{z} + D\boldsymbol{y}) \\ = & \ \boldsymbol{z}^T T_A \boldsymbol{z} + \boldsymbol{y}^T D \boldsymbol{y} \\ \geq & \ 0 \end{aligned} \tag{102}$$

where we have used the fact that T_A and D are symmetric and that V is the kernel of T_A. Now since $\boldsymbol{z}^T T_A \boldsymbol{z}$ and $\boldsymbol{y}^T D \boldsymbol{y}$ vanish only if $\boldsymbol{z}$ and $\boldsymbol{y}$ are zero respectively, we see that the matrix $T_A + D$ is positive definite and hence nonsingular. Moreover, the condition number of the matrix is of order unity. Thus we compute the solution λ of the system

$$(T_A + D)\lambda = b_A \tag{103}$$

The flux splittings are obtained to be

$$\alpha_{KL} = \frac{1}{2} + \frac{\widehat{\mu}_{KL}}{\rho_{KL}} \tag{104}$$

where $\widehat{\mu}$ is obtained from λ using equation (93). Owing to the special structure of M_A, this formula can be simplified further to give

$$\alpha_{KL} = \frac{1}{2} + \frac{\lambda_K - \lambda_L}{\rho_{KL}} \tag{105}$$

The final splitting factor to be used in (68) is given by

$$\alpha_{KL}^{(k)}(s) = \sum_A \alpha_{KL,A}^{(k)} \psi_A(s), \ s \in \Gamma_{KL} \tag{106}$$

Of course, most of the terms in this summation vanish due to ψ_A having non-zero values on a small number of edges. For example, in the case of regular meshes, only two terms in the summation are non-zero, namely those corresponding to the two nodes forming the endpoints of the edge Γ_{KL}. In the case of irregular meshes, the situation is more complicated with at most three non-zero terms appearing in the sum.

Model Problem Example

The following equations were solved on the square domain $\Omega = \left(\frac{1}{2},1\right) \times \left(0,\frac{1}{2}\right)$

$$\begin{aligned} -\ \varepsilon\Delta u &+ x\frac{\partial u}{\partial x} - y\frac{\partial u}{\partial y} + xyu + v = 0 \\ -\ \varepsilon\Delta v &+ x\frac{\partial v}{\partial x} - y\frac{\partial v}{\partial y} - u - xyv = 0 \\ -\ \varepsilon\Delta w &+ x\frac{\partial w}{\partial x} - y\frac{\partial w}{\partial y} = 0 \end{aligned} \tag{107}$$

subject to Dirichlet boundary conditions coinciding with the known analytic solution

$$\begin{aligned} u_0 &= e^{\left(x^2-y^2-1\right)/2\varepsilon} \\ v_0 &= xye^{\left(x^2-y^2-1\right)/2\varepsilon} \\ w_0 &= xy \end{aligned} \tag{108}$$

where ε was chosen to be 0.01.

Estimate Formulation

This problem was solved on a sequence of hp-adapted finite element meshes and the errors in the approximations were estimated using a version of the element residual technique with flux balancing. Since the formulation of the element residual technique is neither obvious nor unique due to the unsymmetric nature of this problem, we provide a brief derivation. First, the variational forms for the governing equations can be written

$$\begin{aligned} B_1\left(\boldsymbol{u},\phi_1\right) &= \int_\Omega \left[\varepsilon\nabla u\cdot\nabla\phi_1 + \left(x\frac{\partial u}{\partial x} - y\frac{\partial u}{\partial y} + xyu + v\right)\phi_1\right] d\Omega = 0 \\ B_2\left(\boldsymbol{u},\phi_2\right) &= \int_\Omega \left[\varepsilon\nabla v\cdot\nabla\phi_2 + \left(x\frac{\partial v}{\partial x} - y\frac{\partial v}{\partial y} - u - xyv\right)\phi_2\right] d\Omega = 0 \\ B_3\left(\boldsymbol{u},\phi_3\right) &= \int_\Omega \left[\varepsilon\nabla w\cdot\nabla\phi_3 + \left(x\frac{\partial w}{\partial x} - y\frac{\partial w}{\partial y}\right)\phi_3\right] d\Omega = 0 \end{aligned} \tag{109}$$

$$\forall\ \boldsymbol{\phi} = (\phi_1,\phi_2,\phi_3) \in \chi = \left[H^1(\Omega)\right]^3$$

where $\boldsymbol{u} = (u,v,w)$. By denoting

$$\mathcal{B}(\boldsymbol{u},\boldsymbol{v}) = \sum_{i=1}^{3} B_i(\boldsymbol{u},\boldsymbol{v}) \tag{110}$$

the problem may be stated:

Find $\boldsymbol{u} \in \chi$ such that

$$\mathcal{B}(\boldsymbol{u}, \boldsymbol{\phi}) = 0 \qquad \forall\, \boldsymbol{\phi} \in \chi \tag{111}$$

The approximate solution comes from a subspace of χ and for this particular example, can be formulated as

Find $\widehat{u} \in \widehat{\chi} \subset \chi$ such that

$$\mathcal{B}\left(\hat{\boldsymbol{u}}, \hat{\boldsymbol{\phi}}\right) = 0 \qquad \forall\, \hat{\boldsymbol{\phi}} \in \widehat{\chi} \tag{112}$$

The error $\boldsymbol{e} = (e_u, e_v, e_w)$ in the corresponding approximation solution $\hat{\boldsymbol{u}} = (\widehat{u}, \widehat{v}, \widehat{w})$ can be written

$$\begin{gathered}\mathcal{B}(\boldsymbol{e}, \boldsymbol{\phi}) = -\mathcal{B}\left(\hat{\boldsymbol{u}}, \boldsymbol{\phi}\right) \\ \forall\, \boldsymbol{\phi} \in \chi\end{gathered} \tag{113}$$

For the purposes of error estimation, however, we define another bilinear form,

$$\overline{\boldsymbol{B}}(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega [\varepsilon \nabla \boldsymbol{u} \cdot \nabla \boldsymbol{v} + \boldsymbol{u} \cdot \boldsymbol{v}] d\Omega \tag{114}$$

Our error estimate, $\overline{\boldsymbol{e}}$, is the solution to

$$\overline{\boldsymbol{B}}\left(\overline{\boldsymbol{e}}, \boldsymbol{\phi}\right) = \mathcal{B}(\boldsymbol{e}, \boldsymbol{\phi}) = -\mathcal{B}\left(\hat{\boldsymbol{u}}, \boldsymbol{\phi}\right) \qquad \forall\, \boldsymbol{\phi} \in \chi \tag{115}$$

We use $\overline{\boldsymbol{B}}$ to define an energy-like norm

$$|||\cdot|||^2 = \overline{\boldsymbol{B}}(\cdot, \cdot) \tag{116}$$

and under standard assumptions it can be shown that the energy-like norm of the estimate is equivalent to the error in the H^1 norm

$$|||\overline{\boldsymbol{e}}||| \sim \|\boldsymbol{e}\|_1 \tag{117}$$

Thus, the global error estimate problem can be written

Find $\overline{\boldsymbol{e}} \in \chi$ such that

$$\overline{\boldsymbol{B}}\left(\overline{\boldsymbol{e}}, \boldsymbol{\phi}\right) = \mathcal{R}\left(\hat{\boldsymbol{u}}, \boldsymbol{\phi}\right) \qquad \forall\, \boldsymbol{\phi} \in \chi \tag{118}$$

where due to the absence of body forces, the residual,

$$\mathcal{R}\left(\hat{\boldsymbol{u}}, \boldsymbol{\phi}\right) \equiv -\mathcal{B}\left(\hat{\boldsymbol{u}}, \boldsymbol{\phi}\right) \tag{119}$$

Finally, the error estimate is

$$\eta = |||\overline{e}||| = \sqrt{\overline{\boldsymbol{B}}\left(\overline{e},\overline{e}\right)} \tag{120}$$

In order to reduce this global problem to element-by-element problems, the constraint of interelement continuity of the error is relaxed and then reapplied using Lagrange multipliers. The Lagrange multiplier is roughly identifiable with the interelement flux which is approximated by the equilibrated fluxes of the approximate solution. Thus, the error estimation problem for an element k can be written

$$\overline{\boldsymbol{B}}_k\left(\overline{e},\phi\right) = \mathcal{R}_k\left(\hat{\boldsymbol{u}},\phi\right) + \int_{\partial K\backslash\partial\Omega} \left\langle \boldsymbol{n}_k \cdot Q^k \right\rangle_{1-\alpha} \phi ds \quad \forall\, \phi \in \chi\left(\Omega_k\right) \tag{121}$$

where the flux term was discussed in this section and the error estimate is given by

$$\eta_k = |||\overline{e}|||_{\partial\Omega_k} = \sqrt{\overline{\boldsymbol{B}}_k\left(\overline{e},\overline{e}\right)} \tag{122}$$

Typically this local problem is solved over some prescribed subset of χ and frequently over a subset of $\chi/\hat{\chi}$ (since $\mathcal{R}(\hat{\boldsymbol{u}},\phi) = 0\ \forall\ \phi \in \hat{\chi}$). We denote this subset as $\mathcal{M}$ and if $\hat{\chi}$ is the set of piecewise polynomials of order $\leq n$, then $\mathcal{M}$ is the set of piecewise polynomials of order m where $n < m \leq n+1$.

Results

This advection diffusion problem was solved on a sequence of hp-adapted meshes. For each mesh the approximate solution, error estimates, and the true errors were calculated. Both the estimate and the true error were evaluated in the energy-like norm defined earlier.

The beginning and final meshes are shown in Figs. 3.4 and 3.7 while gray scale plots of the error estimate and true error are shown in Figs. 3.5 and 3.8 and Figs. 3.6 and 3.9, respectively. These plots are indicative of the overall behavior of this estimate. Namely, the global estimated error is fairly close to the true error (especially for fine meshes) while the distributions of the error estimates are for the most part quite accurate.

Derivation of Error Estimation for Incompressible Flow

The procedure followed in deriving an error estimate for incompressible flows is very similar to the derivation of the estimate presented for the model problem example with two added complications. First, since the governing equations are nonlinear, the formulation of the error is also nonlinear. Unfortunately, this nonlinearity destroys the equivalence between the error estimate and the error. The consequences of this lack of equivalence are not fully understood. In addition, the error in the momentum equation and the error in the incompressibility constraint must be balanced. The procedure

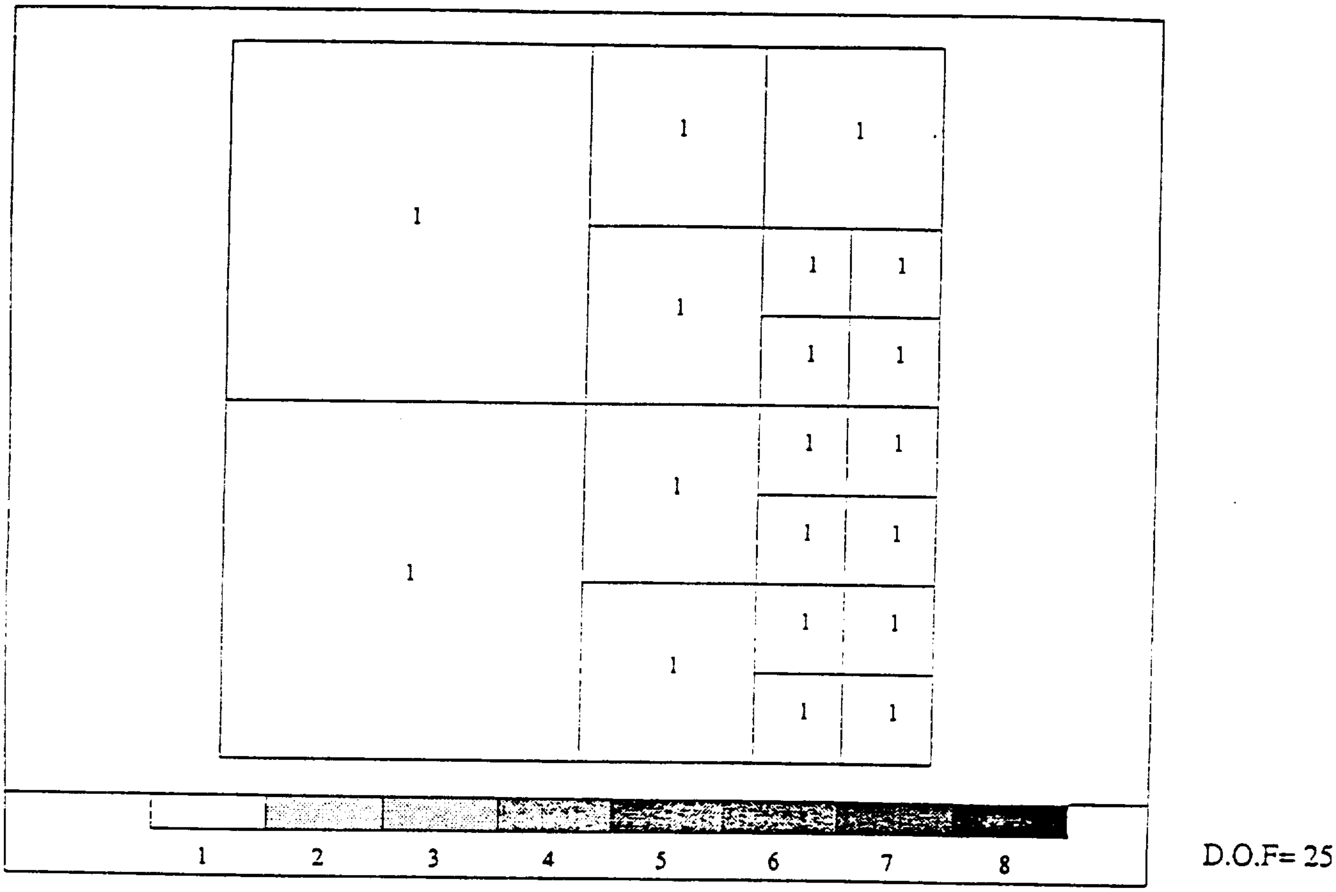
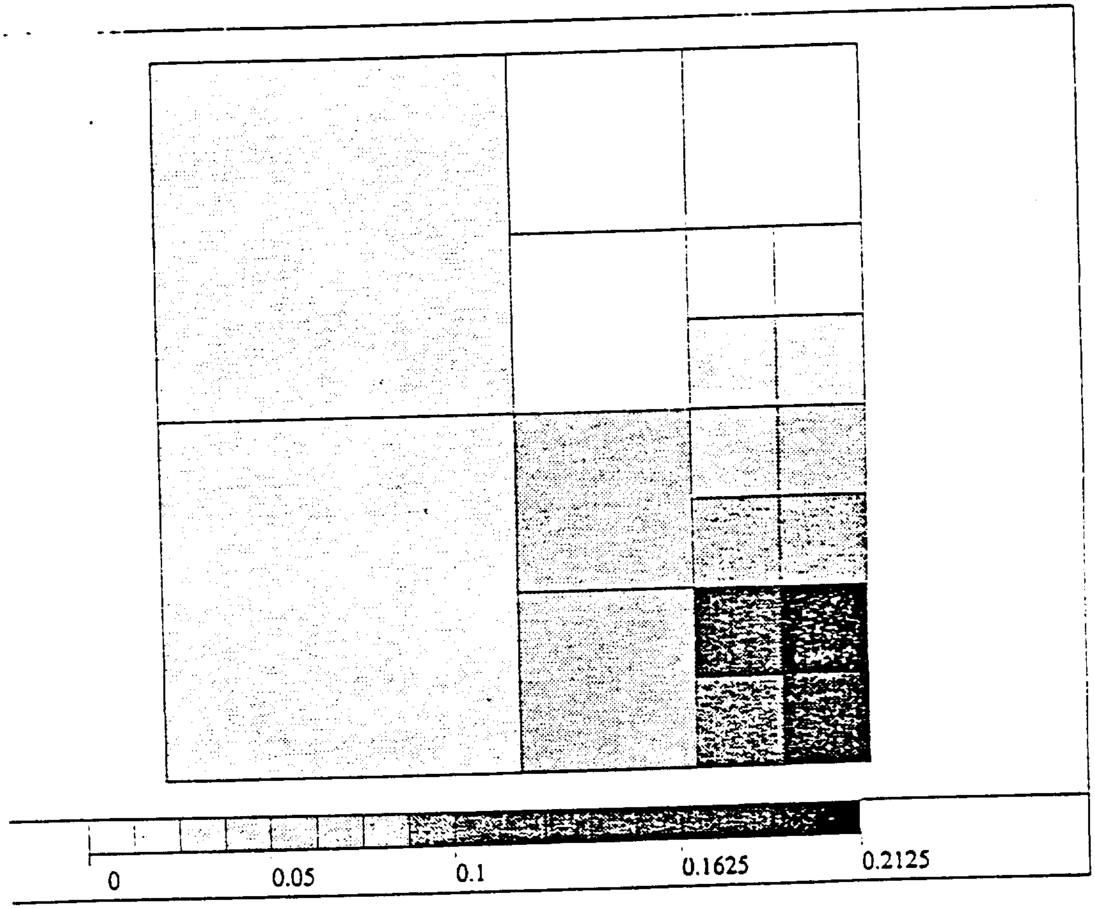


Figure 3.4: Initial mesh used in the model problem example. Mesh consists of 19 bilinear elements.



MIN=0.0034797
MAX=0.204856
ERROR=0.4037959
D.O.F= 25

Figure 3.5: Estimated error distribution on the initial mesh for the model problem example.

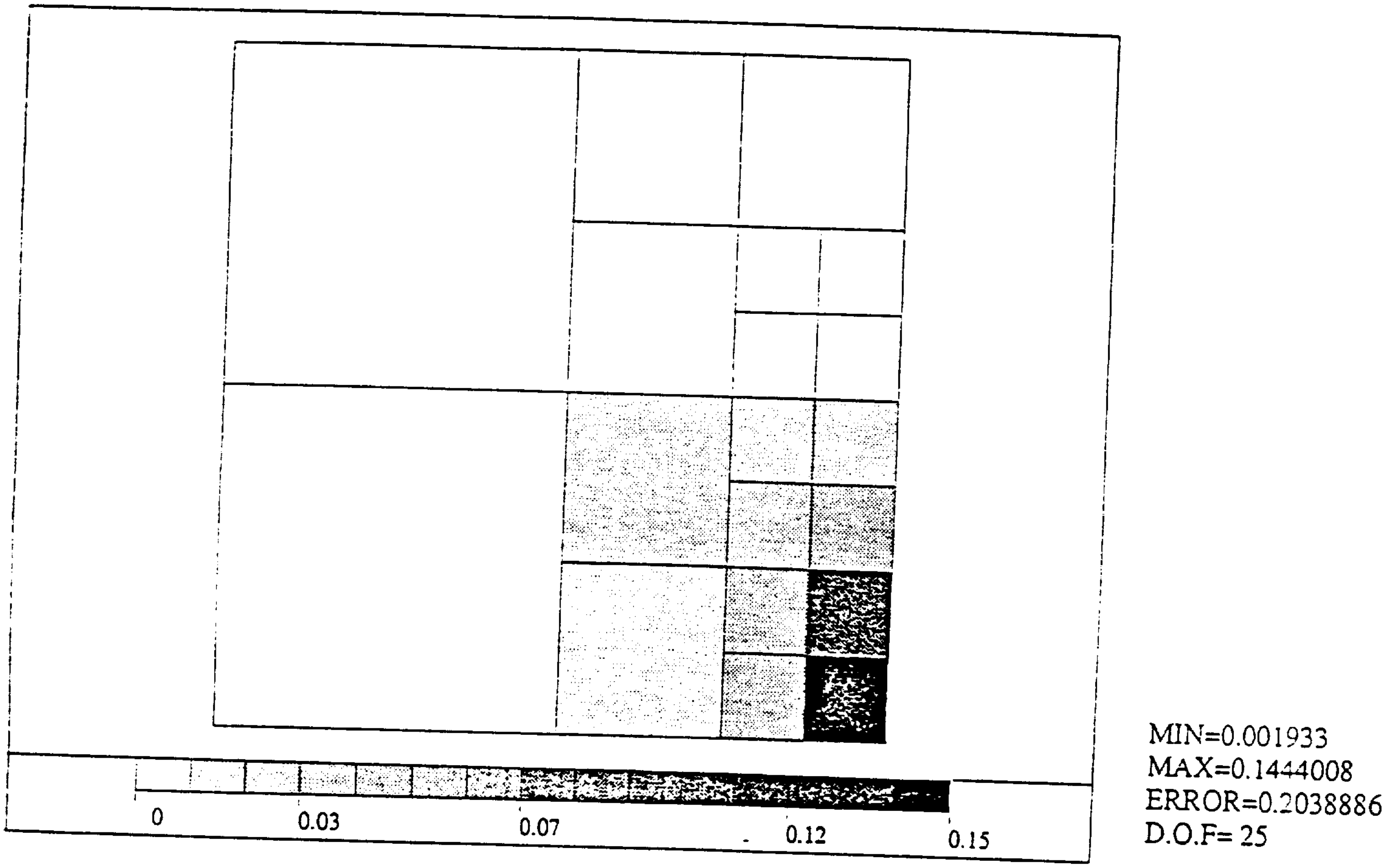


Figure 3.6: Actual error distribution on the initial mesh for the model problem example.

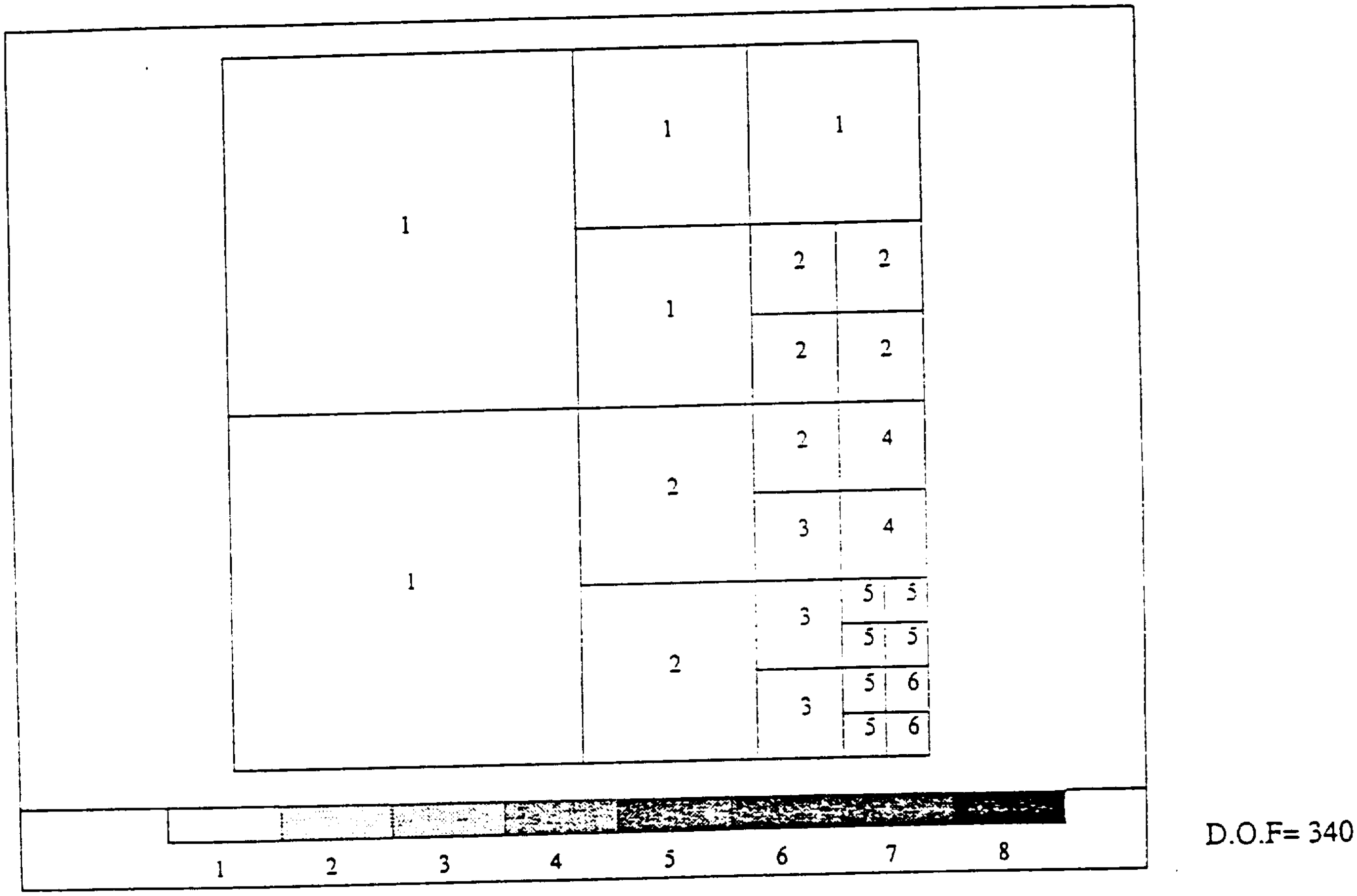
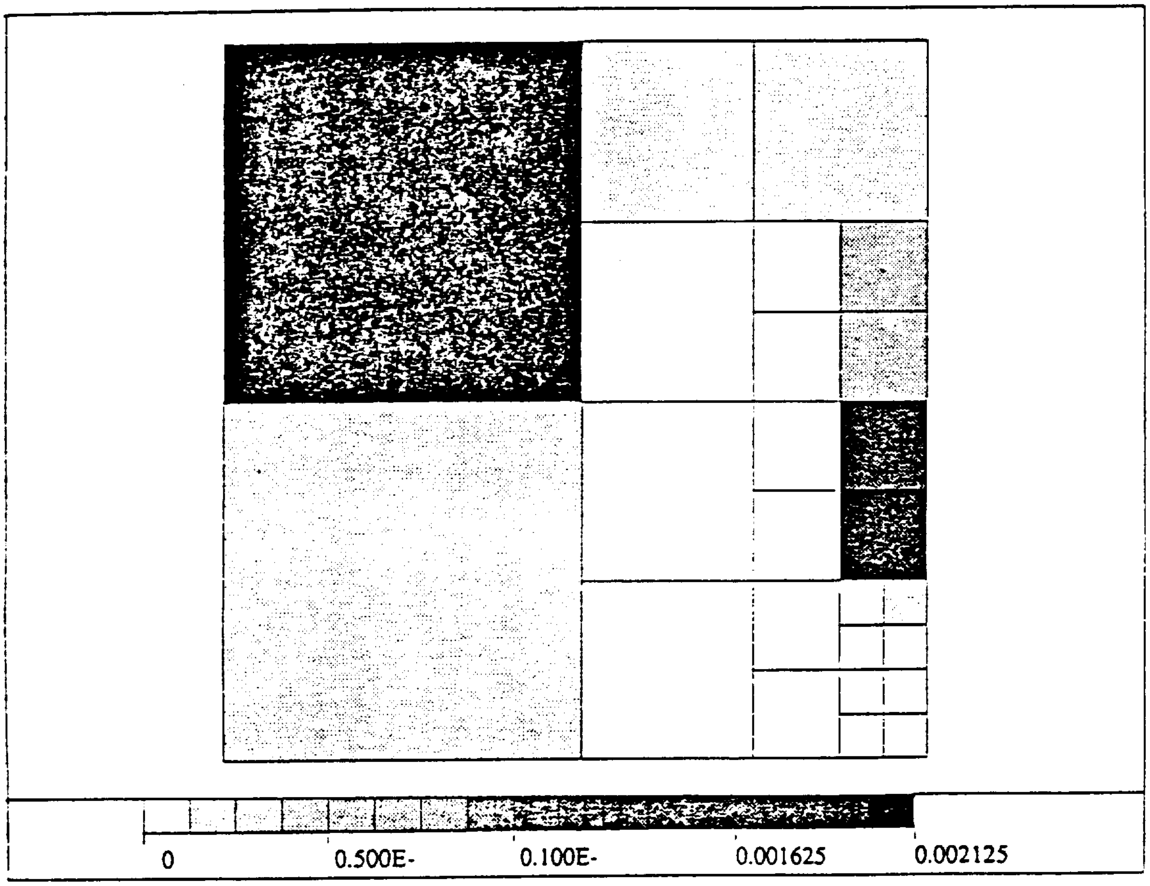


Figure 3.7: Mesh used in the model problem example after four *h*-*p* adaptive passes. Mesh consists of 25 elements ranging from bilinear through sixth polynomial order.



MIN= 0.524E-05
MAX=0.0020944
ERROR=0.0033857
D.O.F= 340

Figure 3.8: Estimated error distribution on the adapted mesh for the model problem example.

by which this was accomplished for Stokes' flow was described in the previous section and a similar technique is used for incompressible flow.

First, the steady state incompressible Navier-Stokes equations are written in the weak form

$$\begin{aligned} &a(\boldsymbol{u},\boldsymbol{v}) + c(\boldsymbol{u},\boldsymbol{u},\boldsymbol{v}) + b(\boldsymbol{v},P) = 0 \\ &b(\boldsymbol{u},q) = 0 \qquad \forall\ (\boldsymbol{v},q) \in \chi \times M \end{aligned} \tag{123}$$

where $\chi = [H^1(\Omega)]^2$ and

$$M = \left\{ q \in L^2(\Omega) \colon \int_\Omega q d\Omega = 0 \right\} \tag{124}$$

and

$$\begin{aligned} a(\boldsymbol{u},\boldsymbol{v}) \quad &= \quad \frac{1}{2}\int_\Omega \nu\left(\nabla\boldsymbol{u} + (\nabla\boldsymbol{u})^T\right) \colon \left(\nabla\boldsymbol{v} + (\nabla\boldsymbol{v})^T\right) d\Omega \\ c(\boldsymbol{w},\boldsymbol{u},\boldsymbol{v}) \quad &= \quad \int_\Omega (\boldsymbol{w}\cdot\nabla)\boldsymbol{u}\cdot\boldsymbol{v}\, d\Omega \\ b(\boldsymbol{v},P) \quad &= \quad -\int_\Omega P(\nabla\cdot\boldsymbol{v}) d\Omega \end{aligned} \tag{125}$$

Taking $\hat{\chi} \subset \chi$ and $\widehat{M} \subset M$ the corresponding approximate solution $(\hat{\boldsymbol{u}},\hat{P}) \in \hat{\chi} \times \widehat{M}$ satisfies

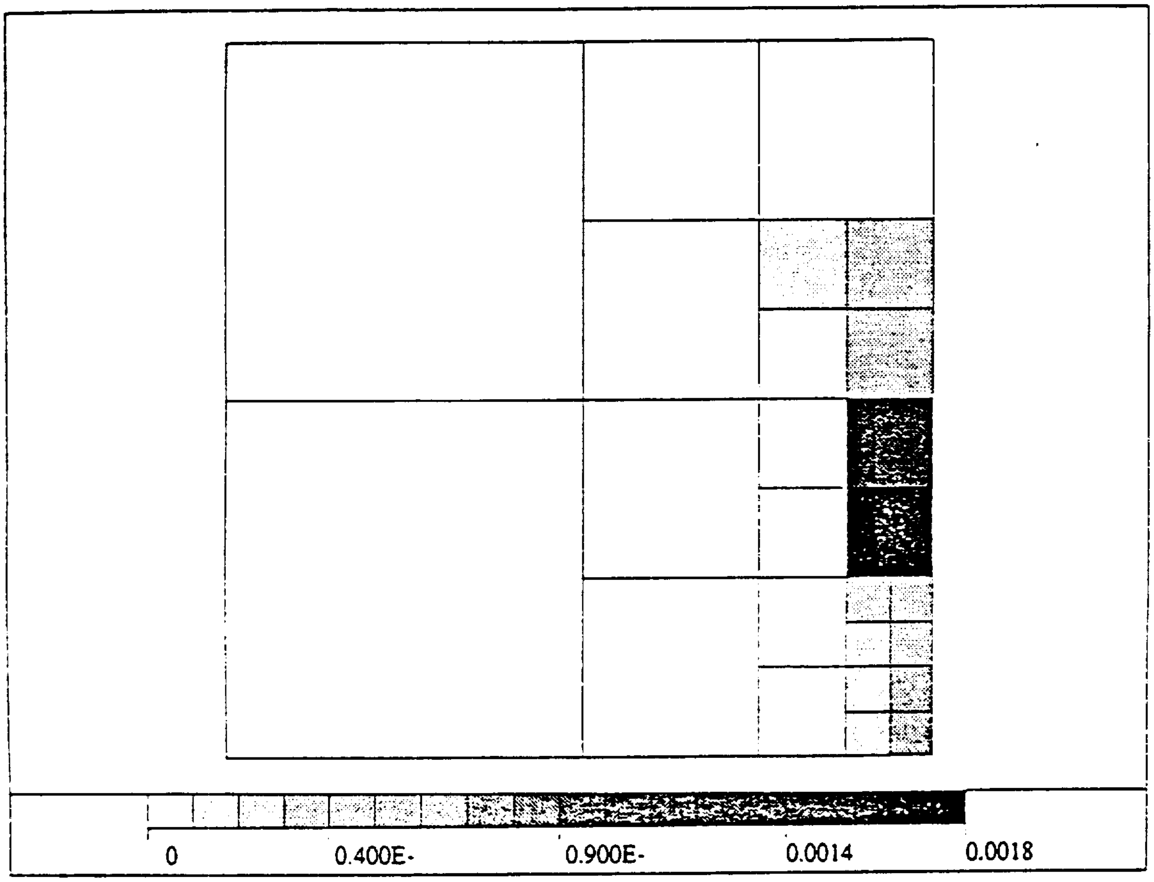
$$\begin{aligned} &a\left(\hat{\boldsymbol{u}},\hat{\boldsymbol{v}}\right) + c\left(\hat{\boldsymbol{u}},\hat{\boldsymbol{u}},\hat{\boldsymbol{v}}\right) + b\left(\hat{\boldsymbol{v}},\hat{P}\right) = 0 \\ &b\left(\hat{\boldsymbol{u}},\hat{q}\right) = 0 \qquad \forall\ \left(\hat{\boldsymbol{v}},\hat{q}\right) \in \hat{\chi} \times \widehat{M} \end{aligned} \tag{126}$$

The formulation for the error, however, differs due to the nonlinear term

$$\begin{aligned} &a(\boldsymbol{e},\boldsymbol{v}) + c(\boldsymbol{e},\boldsymbol{e},\boldsymbol{v}) + c\left(\boldsymbol{e},\hat{\boldsymbol{u}},\boldsymbol{v}\right) \\ &+c\left(\hat{\boldsymbol{u}},\boldsymbol{e},\boldsymbol{v}\right) + b(\boldsymbol{v},E) = \\ &-a\left(\hat{\boldsymbol{u}},\boldsymbol{v}\right) - c\left(\hat{\boldsymbol{u}},\hat{\boldsymbol{u}},\boldsymbol{v}\right) - b\left(\boldsymbol{v},\hat{P}\right) \\ &b(\boldsymbol{e},q) = -b\left(\hat{\boldsymbol{u}},q\right) \end{aligned} \tag{127}$$

where

$$\begin{aligned} \boldsymbol{e} \quad &= \quad \boldsymbol{u} \quad - \quad \hat{\boldsymbol{u}} \\ E \quad &= \quad P \quad - \quad \hat{P} \end{aligned}$$



MIN= 0.659E-05
MAX=0.0017857
ERROR=0.0028596
D.O.F= 340

Figure 3.9: Actual error distribution on the adapted mesh for the model problem example.

Our error estimate $\left(\overline{\boldsymbol{e}}, \overline{E}\right)$ is defined as the solution to

$$\begin{aligned} a\left(\overline{\boldsymbol{e}}, \boldsymbol{v}\right) + \left(\overline{\boldsymbol{e}}, \boldsymbol{v}\right) &= a(\boldsymbol{e}, \boldsymbol{v}) + c(\boldsymbol{e}, \boldsymbol{e}, \boldsymbol{v}) \\ &+ c\left(\hat{\boldsymbol{u}}, \boldsymbol{e}, \boldsymbol{v}\right) + c\left(\boldsymbol{e}, \hat{\boldsymbol{u}}, \boldsymbol{v}\right) \\ &+ b(v, E) \\ \left(\overline{E}, q\right) = b(\boldsymbol{e}, q) & \qquad \forall\ (\boldsymbol{v}, q) \in \chi \times M \end{aligned} \tag{128}$$

where

$$(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega \boldsymbol{u} \cdot \boldsymbol{v}\ d\Omega$$

Correspondingly, we define a weighted H^1 norm

$$|||\boldsymbol{u}||| = \sqrt{a(\boldsymbol{u}, \boldsymbol{u}) + (\boldsymbol{u}, \boldsymbol{u})}$$

At this point we would like to state that the weighted norm of the estimate is equivalent to the H^1 norm of the error but unfortunately this relationship is lost due to the nonlinearity.

For a sufficiently accurate approximation, however, using $\boldsymbol{u} \approx \hat{\boldsymbol{u}}$ and $\boldsymbol{e} \ll \hat{\boldsymbol{u}}$ and we can neglect the nonlinear term and recover the equivalence relationship. With this in mind, the error estimate is formulated

$$\begin{aligned} a\left(\overline{\boldsymbol{e}}, \boldsymbol{v}\right) + \left(\overline{\boldsymbol{e}}, \boldsymbol{v}\right) &= \mathcal{R}_\epsilon\left(\hat{\boldsymbol{u}}, \hat{P}, \boldsymbol{v}\right) \\ \left(\overline{E}, q\right) &= \mathcal{R}_E\left(\hat{\boldsymbol{u}}, q\right) \end{aligned} \tag{129}$$

where the residuals are defined

$$\begin{aligned} \mathcal{R}_e\left(\hat{\boldsymbol{u}}, \hat{P}, \boldsymbol{v}\right) &= -a\left(\hat{\boldsymbol{u}}, \boldsymbol{v}\right) - c\left(\hat{\boldsymbol{u}}, \hat{\boldsymbol{u}}, \boldsymbol{v}\right) - b\left(\boldsymbol{v}, \hat{P}\right) \\ \mathcal{R}_E\left(\hat{\boldsymbol{u}}, q\right) &= -b\left(\hat{\boldsymbol{u}}, q\right) \end{aligned} \tag{130}$$

As with the previous example, these global equations are reduced to element-by-element formulations and solved over some subspace of $\chi/\hat{\chi}$. The resulting estimates are evaluated in terms of their corresponding norms

$$\begin{aligned} \eta_e^k &= |||\overline{\boldsymbol{e}}|||_{\Omega_k} \\ \eta_E^k &= \left\|\overline{E}\right\|_{L_2(\Omega_k)} \end{aligned} \tag{131}$$

Using balancing arguments, these estimates are combined to yield the final error estimate for an element

$$\eta^k = \sqrt{(\eta_\varepsilon^k)^2 + \frac{1}{\varepsilon}\left(\eta_E^k\right)^2} \tag{132}$$

where $\varepsilon > 0$ is a user-specified parameter weighting the two error estimates. For our particular example, ε was assigned the value unity.

Results—Laminar Flow Over a Backstep

This method of error estimation was applied to the solution of laminar flow over a backstep ($Re = 200$) on a mesh of 196 biquadratic elements. The stream function corresponding to this solution is shown in Fig. 3.10 while the estimated errors are shown in Fig. 3.11. For the purposes of comparison, a solution was computed on a mesh of 196 fifth order elements (having over 5000 degrees of freedom) and the difference between this highly accurate solution and the solution on the original quadratic mesh was measured in terms of the weighted H^1 norm of the velocity difference and the L^2 norm of the pressure difference. The resulting plot is shown in Fig. 3.12. Note that the global quantities are within a factor of two of each other and the distributions of error estimates and solution differences are very similar.

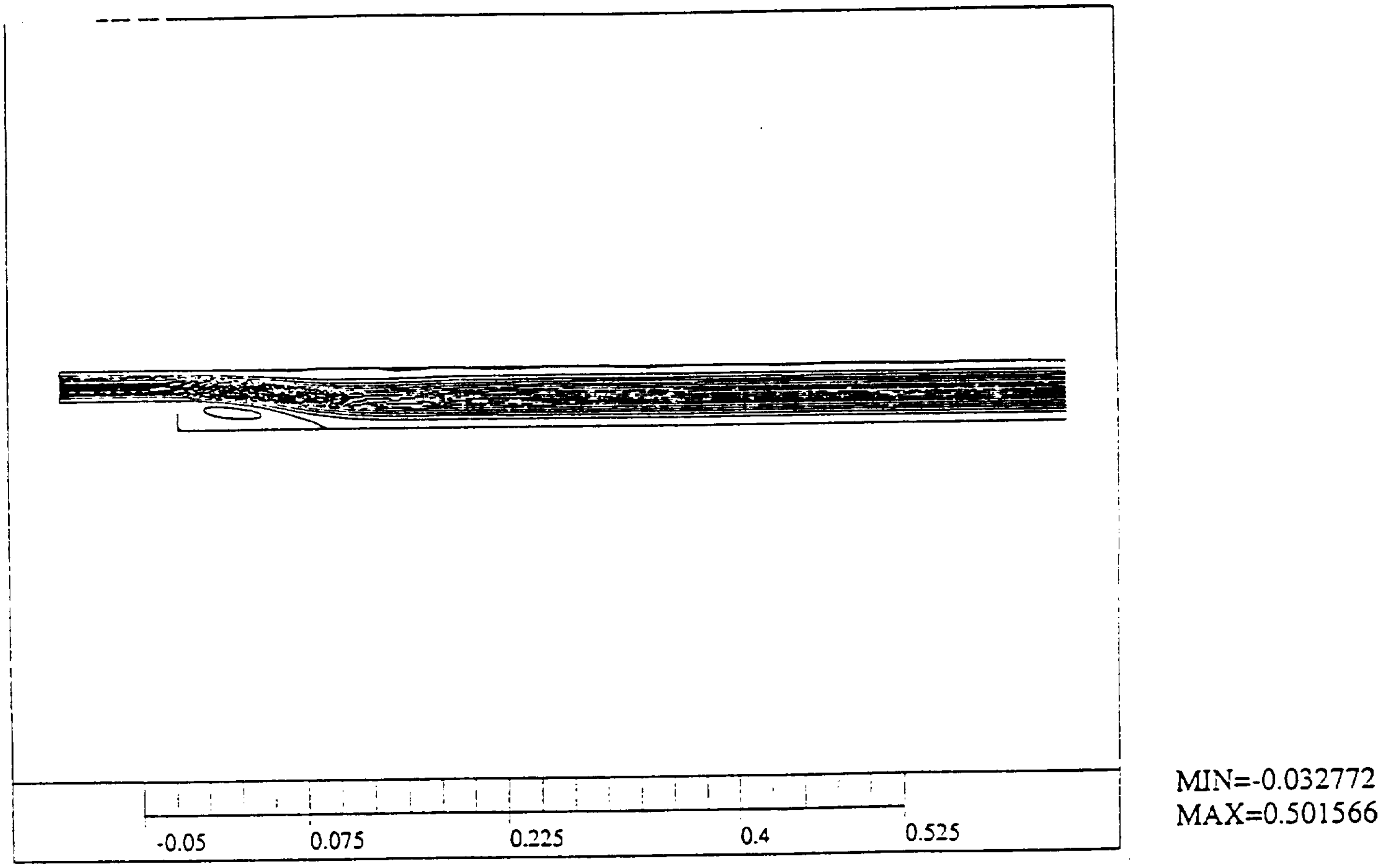


Figure 3.10: Stream function corresponding to solution of laminar flow over a backstep at $Re = 200$ on a mesh of 196 biquadratic elements.

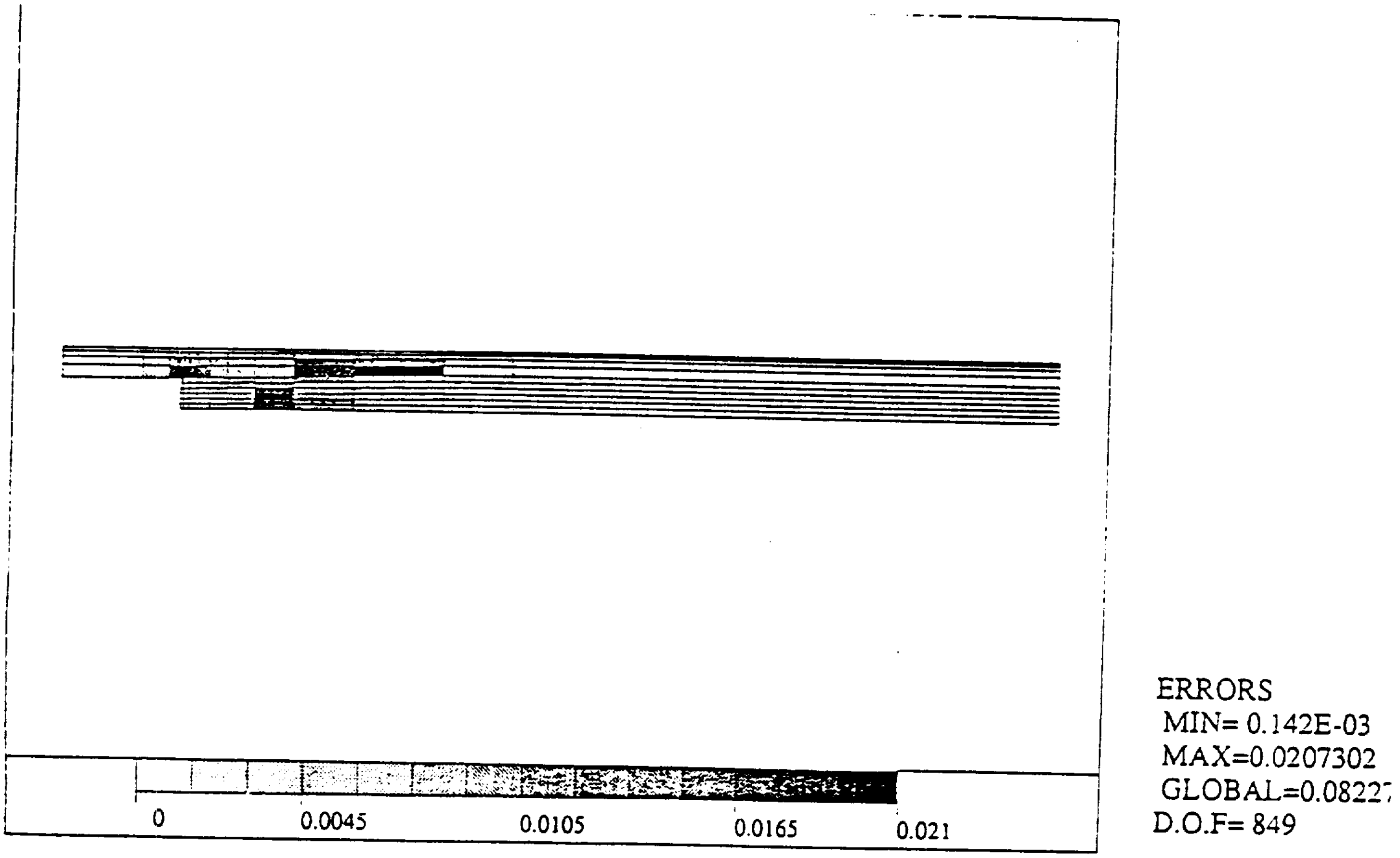


Figure 3.11: Error estimates for solution to laminar flow over a backstep at $Re = 200$ on a mesh of 196 biquadratic elements.

3.2.4 Error Estimation and Postprocessing for Multiple Problem Types

In this section we identify the issues which arise when one attempts to construct a general purpose package which is capable of analyzing the solution obtained when an elliptic boundary value is approximated using a h-p finite element discretization. The aim is to provide techniques which are applicable to a very broad class of problems:

- self-adjoint elliptic systems of partial differential equations (e.g. Lame-Navier equations of linear elasticity)
- problems subject to a constraint (e.g. Stokes' approximation to the incompressible Navier Stokes equations)
- non-self-adjoint and convection dominated problems such as those arising in heat transfer and fluid flow
- problems with strong convection and a constraint (e.g. Oseen's approximation to the incompressible Navier Stokes equations)
- non-linear problems with some or all of the above features (e.g. the incompressible Navier Stokes equations themselves).

The main purpose of the techniques may be divided into two categories:

1. To provide some numerical estimate of the discretization error or quality of the solution (i.e. to obtain *a posteriori error estimates*)
2. To provide enhanced approximations over the original approximation to various derived quantities associated with the solution (e.g. to *postprocess* the solution).

Moreover, the techniques should be able to cope with and perform effectively for general h-p finite element approximations. That is, the underlying mesh used in the discretization may contain irregular or constrained nodes and have elements on which the polynomial degree may vary from element to element. Additionally, the solution may consist of components which vary in degree of approximation and smoothness. For example, in the case of approximation of the Stokes equations one must allow for the pressure approximation to be discontinuous between the elements and also of lower order than the approximation to the velocities.

Lastly, but by no means least, the technique must be reasonably economical. That is to say, the cost of the postprocessing should not exceed the cost of resolving the original problem using a refined discretization.

The first question to be resolved is that of how the error should be assessed (i.e. in what norm it should be measured). Traditionally, the energy norm has always been chosen, mainly

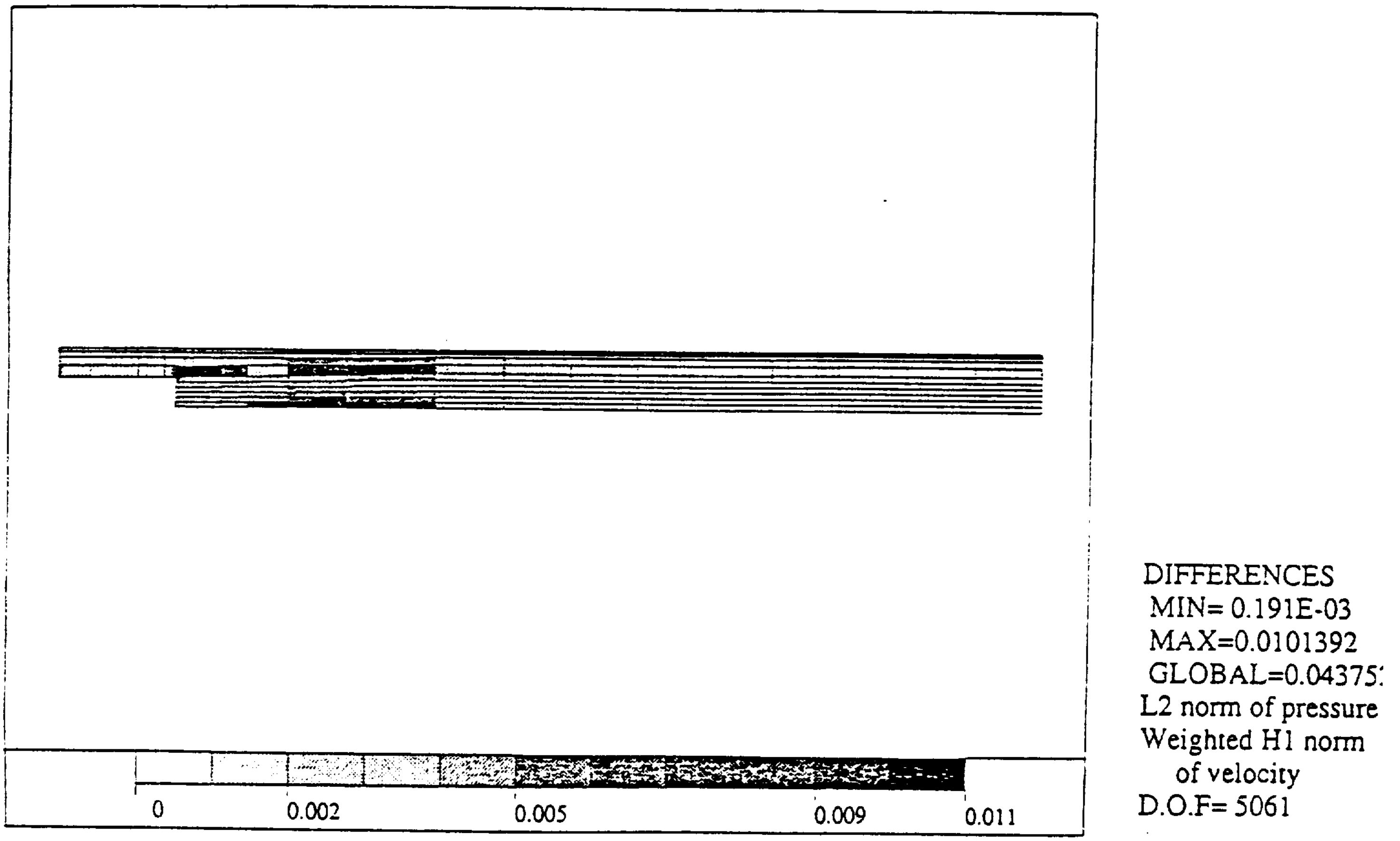


Figure 3.12: Norms of differences between solutions to laminar flow over a backstep at $Re = 200$ from meshes of 196 biquadratic and 196 fifth order elements.

because of the difficulties of working with other norms. While it is convenient to work with the energy norm in the case of self-adjoint problems, for other classes of problem the choice is less clear cut since there is often no natural norm with which to work. Moreover, in the case of problems for which there is a constraint, one has to come to a decision on the correct way in which to deal with this feature.

The question still remains of how the error should be estimated. Currently, the techniques used may be divided into two categories: those based on solving an error residual problem c.f. [10,9] and those based on a postprocessed solution c.f. [15,16]. The approaches which are based on using a postprocesses solution appear at first to be very attractive. Unfortunately, it has been found both in theory and in practice that the techniques perform poorly when either the mesh or the true solution is not sufficiently (very!) smooth. Moreover, as we shall see below, for the problem classes which we have identified above, the correct approach to postprocessing is not at all clear. Conversely, the error residual technique seems to perform reliably even in the cases when the mesh and/or solution are not smooth. However, it is not immediately clear how the approach can be extended to all of the classes we have discussed above.

In summary, the following issues must be addressed when designing a general purpose error estimation package:

- what norm can be used in the case of non-self-adjoint problems
- how should the incompressibility constraint (e.g. in Stokes problem) be dealt with
- what basic approach should be taken (i.e. error residual calculation versus postprocessing).

In this section we shall give a rather complete answer to all of these problems associated with error estimation based on error residual methods. This is particularly pleasing from the point of view of developing a general purpose accuracy assessment package since it reduces the number of basic algorithms which must be present in the code.

The issues associated with solution enhancement are less clear cut than in the case of error estimation. In part, this arises owing to the varied nature of the quantities in which the user is interested. All the user requires of an error estimator is that it provide an estimate of the global error and the local error distribution.

The case of postprocessing is more complicated. For example, while performing an analysis of a linear elastic structure the user may require information on the strain or the stress at a point. Equally well, a derived quantity such as the stress intensity factor near a corner might be of interest. In solving a fluid flow problem, the user is interested in different types of quantities - in this case, primary quantities such as the velocity or pressure profiles. These issues have been addressed to some extent in the literature. For certain types of problems, such as linear elasticity, very effective techniques have been proposed based on the use of extraction formulas [17,18,19]. Meanwhile, for the other classes of problem such techniques are as yet unavailable. It is noteworthy that in neither of the surveys [20,21] is there mention of recovery techniques suitable for the classes of problems described earlier. However, the preprint [22] does contain some ideas specifically tailored to fluid flow problems, but unfortunately does not readily extend to full h-p finite element discretizations.

It can be anticipated that no single technique will provide the best application for every problem class. Thus, one needs to have flexibility to allow the user to input information according to the type of problem being solved and to the type of quantity to be recovered. The actual algorithm to be used would then be tailored based on both of these pieces of information.

In the following sections we deal with each problem class in turn and outline how one can construct effective and reliable *a posteriori* error estimates and also discuss possible solution enhancement techniques. Numerical examples are given illustrating the behavior of the algorithms for various test problems.

Linear Elastostatic Problems

In this section, we discuss *a posteriori* error estimation for linear elastostatic problems. The purpose of this is twofold; first, to show how to obtain accurate *a posteriori* error estimates for *h-p* approximations to such problems, and; second, the theory developed in this section will be crucial in the subsequent sections. The material contained in this section is based on the work presented in [23].

Let us begin by considering a linearly-elastic body occupying an open bounded region $\Omega \subset \mathbb{R}^2$ with boundary $\partial\Omega$. The equations of equilibrium are written in the form

$$\begin{aligned} -\boldsymbol{D}^t\boldsymbol{E}\boldsymbol{D}\boldsymbol{u} &= \boldsymbol{f} \text{ in } \Omega \\ \boldsymbol{u} &= 0 \text{ on } \Gamma_D \\ \boldsymbol{H}(\boldsymbol{n})\boldsymbol{E}\boldsymbol{D}\mathrm{u} &= \boldsymbol{g} \text{ on } \Gamma_N \end{aligned} \tag{133}$$

where $\boldsymbol{D}$ is the divergence operator, $\boldsymbol{E}$ is the elasticity matrix and $\boldsymbol{H}$ is the matrix formed using the components of the unit outward pointing normal vector $\boldsymbol{n} = (n_1, n_2)$ as follows

$$\boldsymbol{H}(\boldsymbol{n}) = \begin{bmatrix} n_1 & 0 & n_2 \\ 0 & n_2 & n_1 \end{bmatrix} \tag{134}$$

and $\boldsymbol{g}, \boldsymbol{f}$ are the applied tractions and body force.

It is convenient to introduce the bilinear form

$$B(\boldsymbol{u}, \boldsymbol{v}) = \int_\Omega (\boldsymbol{D}\boldsymbol{u})^t \boldsymbol{E}(\boldsymbol{D}\boldsymbol{v})dx \tag{135}$$

and the work done by the external forces is defined by the functional

$$L(\boldsymbol{v}) = \int_{\Omega} \boldsymbol{f}^t \boldsymbol{v} dx + \int_{\Gamma_N} \boldsymbol{g}^t \boldsymbol{v} ds \tag{136}$$

With this notation the problem (133) can be rewritten as:

find $\boldsymbol{u} \in X$ such that for all $\boldsymbol{v} \in X$

$$B(\boldsymbol{u}, \boldsymbol{v}) = L(\boldsymbol{v}) \tag{137}$$

The energy norm for this problem is defined by

$$\|\boldsymbol{u}\|_E = \sqrt{B(\boldsymbol{u}, \boldsymbol{u})} \tag{138}$$

The finite element approximation to $\boldsymbol{u}$ is obtained by performing a finite element discretization $\hat{X}$ of the space X and finding $\hat{\boldsymbol{u}} \in \hat{X}$ such that for all $\hat{\boldsymbol{v}} \in \hat{X}$

$$B(\hat{\boldsymbol{u}}, \hat{\boldsymbol{v}}) = L(\hat{\boldsymbol{v}}) \tag{139}$$

The error is defined by $\boldsymbol{e} = \boldsymbol{u} - \hat{\boldsymbol{v}}$. The goal is to construct an *a posteriori* error estimates of the error measured in the energy norm. Choosing $\boldsymbol{v} \in X$ and using (137) gives

$$B(\boldsymbol{e}, \boldsymbol{v}) = L(\boldsymbol{v}) - B(\hat{\boldsymbol{u}}, \boldsymbol{v}) \tag{140}$$

The quantity appearing in the right hand side of (140) contains the residual due to the approximation $\hat{\boldsymbol{u}} \approx \boldsymbol{u}$ in the equations (133).

Local Error Residual Problems

Following the steps in the previous section, we reduce the single global problem stated in (133) into a sequence of local problems. As explained earlier, it is important to perform this decomposition in an appropriate manner if we are to obtain realistic error estimators. Specifically, it was shown how sensitive the effectiveness of the resulting error estimator will be to the way in which the interelement fluxes are decoupled. With this in mind, we introduce a quantity representing an approximation to the normal flux across an interelement boundary as follows

$$\langle \sigma_n(\hat{\boldsymbol{u}}) \rangle \approx \boldsymbol{H}(\boldsymbol{n})\sigma \tag{141}$$

In the next section, we discuss various possible choices of interelement flux. With these considerations in mind, we now decouple (140) in error residual problems over each element K in the mesh:

find $\phi_K \in X_K$ such that for all $\boldsymbol{w} \in X_K$

$$B_K(\phi_K, \boldsymbol{w}) = L_K(\boldsymbol{w}) - B_K(\hat{\boldsymbol{u}}, \boldsymbol{w}) + \int_{\partial\Omega_K \backslash \partial\Omega} \boldsymbol{w}^t \langle \sigma_n(\hat{\boldsymbol{u}}) \rangle ds \tag{142}$$

Mesh	True Error	With Equilibration		Without Equilibration	
		Error	Effectivity	Error	Effectivity
1	0.37113	0.39692	1.06949	0.40592	1.09373
2	0.24601	0.26576	1.08028	0.26792	1.08909
3	0.20319	0.22043	1.08486	0.21823	1.07405
4	0.12711	0.16047	1.26242	0.14935	1.17498
5	0.96581(-1)	0.12593	1.30384	0.10372	1.07397
6	0.56591(-1)	0.88939(-1)	1.57161	0.63038(-1)	1.11392

Table 1: Performance of Error Estimators for short cantilever example

The notation $B_K(\cdot,\cdot)$ and $L_K(\cdot)$ refers to the bilinear and linear forms defined in (135) and (136) evaluated over the subdomain K rather than over the whole domain Ω, while the space X_K refers to the subspace consisting of the restrictions of functions in X to the single subdomain K.

The approach is now to approximate the solution of (142) using a purely local finite element discretization as described in [24]. Having computed ϕ_K, the error on the element K is estimated as ϵ_K where

$$\epsilon_K^2 = B_K(\phi_K, \phi_K) \tag{143}$$

Approximation of Interelement Flux

In order to determine an approximation of the interelement flux, we shall consider two approaches. It is worthwhile to bear in mind that along an interelement edge the finite element approximation of the normal flux is usually discontinuous owing to the discontinuity in the strains. Therefore, along any given edge, there are two possible fluxes corresponding to evaluating the flux using either of the two elements. One approach is to argue that a suitable approximation is to simply average the two contributions along an edge between the elements forming the edge. An alternative approach is to try to perform a weighted averaging of the fluxes and hopefully achieve an improvement over the simple averaging. Further details concerning the second approach will be found in [24,25]. In particular, for the latter scheme is is possible to show [10], that the error estimator which one obtains *provides a guaranteed upper bound on the true discretization error.*

Numerical Examples

In order to test the behavior of the error estimators described above, we apply each one to various test problems.

Example: Short Cantilever

The geometry of the problem is shown in Figure 3.13.

Mesh	True	With Equilibration		Without Equilibration	
	Error	Error	Effectivity	Error	Effectivity
1	0.53505	0.60633	1.13323	0.59951	1.12010
2	0.31535	0.41129	1.30420	0.34988	1.10949
3	0.28700	0.38627	1.34588	0.32205	1.12213
4	0.77801(-1)	0.11322	1.45510	0.12039	1.54723

Table 2: Performance of Error Estimators for cracked panel example

Figures 3.14 - 3.19 show the plots of the meshes used for computation and the resulting distributions of error estimates, while Table 1 lists the true and estimated (global) errors. Observe that as predicted theoretically, the estimates do provide an upper bound on the error.

Example: Cracked Panel

The geometry of the problem is shown in Figure 3.20

This time we assume plane strain conditions and Poisson's ratio of 0.3. The results of the adaptive analysis are shown in Table 2, while the sequence of meshes and corresponding error distributions are shown in Figures 3.21 - 3.24. Due to the symmetry of the problem it is only necessary to solve for half of the panel, which is loaded so that there is a stress singularity of the form $r^{-\frac{1}{2}}$ near the crack tip. As with the previous example, the results confirm that the error estimate provides an upper bound on the actual discretization error.

Stokes' and Oseen's Equations

In this section we consider the following model incompressible problems:

- Stokes equations

$$\begin{aligned} -\nu\Delta\boldsymbol{u} + \nabla p &= \boldsymbol{f} \\ \nabla\cdot\boldsymbol{u} &= \boldsymbol{0} \end{aligned} \tag{144}$$

- Oseen's equations

$$\begin{aligned} -\nu\Delta\boldsymbol{u} + (\boldsymbol{b}\cdot\nabla)\boldsymbol{u} + \nabla p &= \boldsymbol{f} \\ \nabla\cdot\boldsymbol{u} &= \boldsymbol{0} \end{aligned} \tag{145}$$

subject to appropriate boundary conditions. It will be convenient to write each of these problems in a generic weak form as follows: find $\boldsymbol{u}, p$ such that for all $\boldsymbol{v}$ and q there holds

$$\begin{aligned} B(\boldsymbol{u},\boldsymbol{v}) + b(\boldsymbol{v},p) &= L(\boldsymbol{v}) \\ b(\boldsymbol{u},q) &= 0 \end{aligned} \qquad (146)$$

Remark: Strictly speaking the above formulation leaves much to be desired in terms of mathematical rigor, but our purposes and for ease of explanation it will suffice.

These problems are natural testing grounds on the way to producing an algorithm suitable for the incompressible Navier Stokes equations. The main features introduced when considering these problems are

- The question of what norm is appropriate to estimate the error for problems with an incompressibility constraint.
- The finite element approximation to the pressure p may be discontinuous and of a different polynomial degree to the approximation to the velocity $\boldsymbol{u}$.
- The issue of how one should deal with the incompressibility constraint. Previously we have seen that it is appropriate to solve an error residual problem for each component of the solution, but it is not obvious that this is the correct direction in the case of a problem containing a constraint.
- The introduction of a convective term in Oseen's equations renders the problem severely non-self adjoint. It is unclear not only in which norm one should estimate the error, but also whether the convective effect will cause the local distribution of the error estimator to be faithful to the actual error distribution.

Symmetrized Norms

In this section we consider the question of the choice of norm in which to measure the error. Letting $\hat{\boldsymbol{u}}$ and p denote the h-p finite element approximation, we define the error in the velocity and the pressure by

$$\begin{aligned} \boldsymbol{e} &= \boldsymbol{u} - \hat{\boldsymbol{u}} \\ E &= p - \hat{p} \end{aligned} \qquad (147)$$

One obvious choice of norm is given by

$$\|(\boldsymbol{e},E)\|_{*} \equiv (\|e\|^2_{H^1} + \|E\|^2_{L_2})^{\frac{1}{2}} \qquad (148)$$

Here H^1 and L_2 denote the standard Sobolev spaces. In essence, the above norm weights the error in the gradients of the velocity against the error in the pressure. While this weighting is appropriate mathematically, it suffers the drawback that it is not readily accessible. In short, while it is desirable to use the star norm (148) it is not computable.

An alternative approach is required and one which suggests itself is to extend the concept of a *symmetrizer* which was successfully used in [26]. The idea which we present below extends the approach in [24,27] to problems involving a constraint.

The first step is to introduce a new pair of bilinear forms

$$\begin{aligned} A(\boldsymbol{u},\boldsymbol{v}) &= \int_\Omega \nu(u_{i,j}+u_{j,i})v_{i,j}dx \\ (p,q) &= \int_\Omega pq dx \end{aligned} \qquad (149)$$

and then to define the pair ϕ,ψ to be solutions of the problems

$$\begin{aligned} A(\phi.\boldsymbol{v}) &= B(\boldsymbol{e},\boldsymbol{v})+b(\boldsymbol{v},E) \\ (\psi,q) &= b(\boldsymbol{e},q) \end{aligned} \qquad (150)$$

and then to consider the quantity

$$\|(\phi,\psi)\| = (A(\phi,\phi)+(\psi,\psi))^{\frac{1}{2}} \qquad (151)$$

The quantity defined in (151) is referred to as the error measured in the *symmetrized* norm. It is easy to suspect that this quantity is related to the true error and this in indeed the case. The following result shows the precise relationship.

Proposition 1 *Suppose that the bilinear forms appearing in (146) are continuous and satisfy an inf-sup condition. Then there are constants $C_1 > 0$ and $C_2 > 0$ for which*

$$C_1\|(\phi,\psi)\| \leq \|(\boldsymbol{e},E)\|_* \leq C_2\|(\phi,\psi)\| \qquad (152)$$

This result shows that while we are unable to make use of the start norm (148) directly the symmetrized error norm (151) is just as good. The next step is to show how we can estimate the error in the symmetrized norm.

Error Estimation in Symmetrized Norms

Examining the (150) and (151) it is easily seen that the error in the symmetrized norm can be rewritten as

$$\|(\phi,\psi)\|^2 = A(\phi,\phi)+\int_\Omega(\text{div }\hat{\boldsymbol{u}})^2dx \qquad (153)$$

This shows that if we want to estimate the error in the symmetrized norm, then it is only necessary to estimate the term $A(\phi,\phi)$. Returning to (150) and using (146) and (147) we obtain

Mesh		Error Estimate		True
h^{-1}	p	Without Balancing	With Balancing	Error
4	2	1.901(-3)	1.778(-3)	1.722(-3)
16	2	4.64(-4)	4.49(-4)	4.15(-3)
4	3	5.232(-5)	5.229(-5)	5.226(-5)
16	3	6.528(-6)	6.528(-6)	6.524(-6)

Table 3: Performance of Error Estimators for smooth Stokes example

$$\begin{aligned} A(\phi, \boldsymbol{v}) &= B(\boldsymbol{e}, \boldsymbol{v}) + b(\boldsymbol{v}, E) \\ &= L(\boldsymbol{v}) - B(\hat{\boldsymbol{u}}, \boldsymbol{v}) - b(\boldsymbol{v}, \hat{p}) \end{aligned} \tag{154}$$

The right hand of (154) is actually computable and represents the residual in the momentum equations from the original problem. A moment's reflection on this equation reveals that the problem determining ϕ is in fact a linear elastic problem owing to the definition of the bilinear form $A(\cdot, \cdot)$. Having reached this stage, one can now compare problem (154) with the analogous problem (142) from the previous section. That is to say, one proceeds exactly as before to define local error residual problems for ϕ. These can then be solved in the same manner yielding the first term in (153) and consequently an estimate of the error.

Reliability of the Error Estimator

The foregoing manipulations show how an *a posteriori* error estimate can be obtained by solving local linear elastic problems similar to those in the previous section. The question which naturally arises is that of how accurate are the error estimates. The following result quantifies the behavior

Proposition 2 *Let $\hat{\phi}$ denote the solution of the local error residual problems derived from (3.22) and define the local error estimate ϵ_K on element K by*

$$\epsilon_K^2 = A_K(\hat{\phi}, \hat{\phi}) + \int_K (\text{div } \hat{\boldsymbol{u}})^2 dx \tag{155}$$

Then

$$\|(\boldsymbol{e}, E)\|_* \leq \frac{1}{C_1} \left\{ \sum_K \epsilon_K^2 \right\}^{\frac{1}{2}} \tag{156}$$

The result shows that the estimator should bound the true error.

Numerical Examples